INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team GRAAL

Algorithms and Scheduling for Distributed Heterogeneous Platforms

Grenoble - Rhône-Alpes

Theme : Distributed and High Performance Computing

Table of contents

The GRAAL *project-team is common to CNRS, ENS Lyon, and INRIA. This project-team is part of the Laboratoire de l'Informatique du Parallélisme (LIP), UMR ENS Lyon/CNRS/INRIA/UCBL 5668. This project-team is located in part at the École normale supérieure de Lyon and in part at the Université Claude Bernard – Lyon 1.*

1. Team

Research Scientist

Frédéric Desprez [Research Director (DR), HdR]
Gilles Fedak [Research Associate (CR)]
Jean-Yves L'Excellent [Research Associate (CR), Acted as Team Leader during Frédéric Vivien's sabbatical]
Loris Marchal [Research Associate (CR)]
Christian Pérez [Research Associate (CR), HdR]
Bora Uçar [Research Associate (CR)]
Frédéric Vivien [Team Leader, Research Associate (CR), HdR]

Faculty Member

Anne Benoît [Assistant Professor (MCF), HdR]
Hinde Bouziane [ATER, until August 31, 2009]
Yves Caniou [Assistant Professor (MCF)]
Eddy Caron [Assistant Professor (MCF)]
Gaël Le Mahec [ATER, until August 31, 2009]
Bernard Tourancheau [Professor, HdR]
Yves Robert [Professor, HdR]

External Collaborator

Sékou Diakité [PhD student, MENRT grant]
Alexandru Dobrila [PhD student, MENRT grant]
Jean-Marc Nicod [Assistant Professor, HdR]
Laurent Philippe [Professor, HdR]
Lamiel Toch [PhD student, MENRT grant]

Technical Staff

Nicolas Bard [CNRS]
Aurélien Ceyden [ENS Lyon, 50% on the project, until August 31, 2009]
Florent Chuffart [INRIA, since September 1, 2009]
Haiwu He [INRIA]
Benjamin Isnard [INRIA]
Guillaume Joslin [INRIA, since October 12, 2009]
Gaël Le Mahec [Since September 1, 2009]
David Loureiro [Since April 15, 2009]
Vincent Pichon [ENS Lyon, until April 5, 2009]
Daouda Traore [INRIA, since October 15, 2009]

PhD Student

Leila Ben Saad [MENRT grant]
Julien Bigot [MENRT grant]
Raphaël Bolze [ENS-AFM grant until January 31, 2009]
Ghislain Charrier [INRIA Cordi-S grant]
Benjamin Depardon [MENRT grant]
Fanny Dufossé [ENS grant]
Matthieu Gallet [ENS grant]
Jean-Sébastien Gay [Rhône-Alpes region grant, on long term leave for health reasons]
Cristian Klein [INRIA grant, since October 1, 2009]

Mathias Jacquelin [MENRT grant]
George Markomanolis [INRIA Cordi-S grant, since December 1, 2009]
Vincent Pichon [CIFRE EDF R&D grant, since April 6, 2009]
Georges Markomanolis [INRIA Cordi-S grant, since December 1, 2009]
Adrian Muresan [MENRT grant, since October 1, 2009]
Veronika Rehn-Sonigo [MENRT grant, until September 30, 2009]
Paul Renaud-Goud [MENRT grant]
Clément Rezvoy [MENRT grant]

Post-Doctoral Fellow

Hinde Bouziane [Since September 1, 2009]
Laurent Bobelin [Since March 9, 2009]
Indranil Chowdhury [Since May 21, 2009]
Michaël Heymann [Since January 10, 2009]

Visiting Scientist

Hidemoto Nakada [AIST,Japan, until February 23, 2009]
Franck Petit [On leave from University of Picardie until August 31, 2009]
Bing Tang [Wuhan University of technology, until October 27, 2009]
Wang Yu [Hohai University, China, until December 6, 2009]

Administrative Assistant

Evelyne Blesle [INRIA, 50% on the project]

2. Overall Objectives

2.1. Introduction

Parallel computing has spread into all fields of applications, from classical simulation of mechanical systems or weather forecast to databases, video-on-demand servers or search tools like Google. From the architectural point of view, parallel machines have evolved from large homogeneous machines to clusters of PCs (with sometime boards of several processors sharing a common memory, these boards being connected by high speed networks like Myrinet). However the need of computing or storage resources has continued to grow leading to the need of resource aggregation through Local Area Networks (LAN) or even Wide Area Networks (WAN). The recent progress of network technology has enabled the use of highly distributed platforms as a single parallel resource. This has been called Metacomputing or more recently Grid Computing [98]. An enormous amount of financing has recently been put on this important subject, leading to an exponential growth of the number of projects, most of them focusing on low level software detail. We believe that many of these projects failed to study fundamental issues such as the computational complexity of problems and algorithms and heuristics for scheduling problems. Also they usually have not validated their theoretical results on available software platforms.

From the architectural point of view, Grid Computing has different scales but is always highly heterogeneous and hierarchical. At a very large scale, tens of thousands of PCs connected through the Internet are aggregated to solve very large applications. This form of the Grid, usually called a Peer-to-Peer (P2P) system, has several incarnations, such as SETI@home, Gnutella or XtremWeb [107]. It is already used to solve large problems (or to share files) on PCs across the world. However, as today's network capacity is still low, the applications supported by such systems are usually embarrassingly parallel. Another large-scale example is TeraGRID which connects several supercomputing centers in the USA and reaches a peak performance of over 100 Teraflops. At a smaller scale but with a high bandwidth, one can mention the Grid'5000 project, which connects PC clusters spread in nine French university research centers. Many such projects exist over the world that connect a small set of machines through a fast network. Finally, at a research laboratory level, one can build an heterogeneous platform by connecting several clusters using a fast network such as Myrinet.

The common problem of all these platforms is not the hardware (these machines are already connected to the Internet) but the software (from the operating system to the algorithmic design). Indeed, the computers connected are usually highly heterogeneous (from clusters of SMPs to the Grid).

There are two main challenges for the widespread use of Grid platforms: the development of environments that will ease the use of the Grid (in a seamless way) and the design and evaluation of new algorithmic approaches for applications using such platforms. Environments used on the Grid include operating systems, languages, libraries, and middlewares [96], [98], [102]. Today's environments are based either on the adaptation of "classical" parallel environments or on the development of toolboxes based on Web Services.

Aims of the GRAAL project.

In the GRAAL project we work on the following research topics:

- algorithms and scheduling strategies for heterogeneous platforms and the Grid,
- environments and tools for the deployment of applications in a client-server mode.

The main keywords of the GRAAL project:
Algorithmic Design + Middleware/Libraries + Applications
over Heterogeneous Architectures and the Grid

2.2. Highlights of the year

- Anne Benoit was elected as a junior member of the Institut Universitaire de France.
- The SysFera startup company created by Eddy Caron, Frédéric Desprez and David Loureiro to transfer the DIET middleware has been awarded the OSEO "*concours national à la création d'entreprise innovante*" 2009 price.

3. Scientific Foundations

3.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Leila Ben Saad, Sékou Diakité, Alexandru Dobrila, Fanny Dufossé, Matthieu Gallet, Mathias Jacquelin, Loris Marchal, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Paul Renaud-Goud, Clément Rezvoy, Yves Robert, Bernard Tourancheau, Frédéric Vivien.

Scheduling sets of computational tasks on distributed platforms is a key issue but a difficult problem. Although a large number of scheduling techniques and heuristics have been presented in the literature, most of them target only homogeneous resources. However, future computing systems, such as the computational Grid, are most likely to be widely distributed and strongly heterogeneous. Therefore, we consider the impact of heterogeneity on the design and analysis of scheduling techniques: how to enhance these techniques to efficiently address heterogeneous distributed platforms?

The traditional objective of scheduling algorithms is the following: given a task graph and a set of computing resources, or *processors*, map the tasks onto the processors, and order the execution of the tasks so that: (i) the task precedence constraints are satisfied; (ii) the resource constraints are satisfied; and (iii) a minimum schedule length is achieved. Task graph scheduling is usually studied using the so-called *macro-dataflow* model, which is widely used in the scheduling literature: see the survey papers [97], [106], [114], [116] and the references therein. This model was introduced for homogeneous processors, and has been (straightforwardly) extended to heterogeneous computing resources. In a word, there is a limited number of computing resources, or processors, to execute the tasks. Communication delays are taken into account as follows: let task T be a predecessor of task T' in the task graph; if both tasks are assigned to the same processor, no communication overhead is incurred, the execution of T' can start immediately at the end of the execution of T; on the contrary,

if T and T' are assigned to two different processors P_i and P_j, a communication delay is incurred. More precisely, if P_i completes the execution of T at time-step t, then P_j cannot start the execution of T' before time-step $t + \text{comm}(T, T', P_i, P_j)$, where $\text{comm}(T, T', P_i, P_j)$ is the communication delay, which depends upon both tasks T and T', and both processors P_i and P_j. Because memory accesses are typically several orders of magnitude cheaper than inter-processor communications, it is sensible to neglect them when T and T' are assigned to the same processor.

The major flaw of the macro-dataflow model is that communication resources are not limited in this model. Firstly, a processor can send (or receive) any number of messages in parallel, hence an unlimited number of communication ports is assumed (this explains the name *macro-dataflow* for the model). Secondly, the number of messages that can simultaneously circulate between processors is not bounded, hence an unlimited number of communications can simultaneously occur on a given link. In other words, the communication network is assumed to be contention-free, which of course is not realistic as soon as the number of processors exceeds a few units.

The general scheduling problem is far more complex than the traditional objective in the *macro-dataflow* model. Indeed, the nature of the scheduling problem depends on the type of tasks to be scheduled, on the platform architecture, and on the aim of the scheduling policy. The tasks may be independent (e.g., they represent jobs submitted by different users to a same system, or they represent occurrences of the same program run on independent inputs), or the tasks may be dependent (e.g., they represent the different phases of a same processing and they form a task graph). The platform may or may not have a hierarchical architecture (clusters of clusters vs. a single cluster), it may or may not be dedicated. Resources may be added to or may disappear from the platform at any time, or the platform may have a stable composition. The processing units may have the same characteristics (e.g., computational power, amount of memory, multi-port or only single-port communications support, etc.) or not. The communication links may have the same characteristics (e.g., bandwidths, latency, routing policy, etc.) or not. The aim of the scheduling policy can be to minimize the overall execution time (makespan minimization), the throughput of processed tasks, etc. Finally, the set of all tasks to be scheduled may be known from the beginning, or new tasks may arrive all along the execution of the system (on-line scheduling).

In the GRAAL project, we investigate scheduling problems that are of practical interest in the context of large-scale distributed platforms. We assess the impact of the heterogeneity and volatility of the resources onto the scheduling strategies.

3.2. Scheduling for Parallel Sparse Direct Solvers

Participants: Guillaume Joslin, Maurice Brémond, Indranil Chowdhury, Jean-Yves L'Excellent, Bora Uçar.

The solution of sparse systems of linear equations (symmetric or unsymmetric, most often with an irregular structure) is at the heart of many scientific applications arising in various domains such as geophysics, chemistry, electromagnetism, structural optimization, and computational fluid dynamics. The importance and diversity of the fields of application are our main motivation to pursue research on sparse linear solvers. Furthermore, in order to solve hard problems that result from ever-increasing demand for accuracy in simulations, special attention must be paid to both memory usage and execution time on the most powerful parallel platforms (whose usage is necessary because of the volume of data and amount of computation induced). This is done by specific algorithmic choices and scheduling techniques. From a complementary point of view, it is also necessary to be aware of the functionality requirements from the applications and from the users, so that robust solutions can be proposed for a large range of problems.

Because of their efficiency and robustness, direct methods (based on Gaussian elimination) are methods of choice to solve these types of problems. In this context, we are particularly interested in the multifrontal method [104], [105] for symmetric positive definite, general symmetric or unsymmetric problems, with numerical pivoting in order to ensure numerical accuracy. The existence of numerical pivoting induces dynamic updates in the data structures where the updates are not predictable with a static or symbolic analysis approach.

The multifrontal method is based on an elimination tree [110] which results (i) from the graph structure corresponding to the nonzero pattern of the problem to be solved, and (ii) from the order in which variables are eliminated. This tree provides the dependency graph of the computations and is exploited to define tasks that may be executed in parallel. In this method, each node of the tree corresponds to a task (itself can be potentially parallel) that consists in the partial factorization of a dense matrix. This approach allows for a good locality and hence efficient use of cache memories.

We are especially interested in approaches that are intrinsically dynamic and asynchronous [1], [100], as these approaches can encapsulate numerical pivoting and can be adopted to various computer architectures. In addition to their numerical robustness, the algorithms are based on a dynamic and distributed management of the computational tasks, not so far from today's peer-to-peer approaches: each process is responsible for providing work to some other processes and at the same time it acts as a worker for others. These algorithms are very interesting from the point of view of parallelism and in particular for the study of mapping and scheduling strategies for the following reasons:

- the associated task graphs are very irregular and can vary dynamically,
- they are currently used inside industrial applications, and
- the evolution of high performance platforms, to the more heterogeneous and less predictable ones, requires that applications adapt themselves, using a mixture of dynamic and static approaches, as our approach allows.

Our research in this field is strongly linked to the software package MUMPS (see Section 5.2) which is our main platform to experiment and validate new ideas and pursue new research directions. We are facing new challenges for very large problems (tens to hundreds of millions of equations) that occur nowadays in various application fields: in that case, either parallel out-of-core approaches are required, or direct solvers should be combined with iterative schemes, leading to hybrid direct-iterative methods.

3.3. Providing Access to HPC Servers on the Grid

Participants: Nicolas Bard, Julien Bigot, Raphaël Bolze, Julien Bigot, Hinde Bouziane, Yves Caniou, Eddy Caron, Aurélien Ceyden, Ghislain Charrier, Florent Chuffart, Benjamin Depardon, Frédéric Desprez, Gilles Fedak, Jean-Sébastien Gay, Haiwu He, Cristian Klein, David Loureiro, Christian Pérez, Vincent Pichon, Bing Tang.

Resource management is one of the key issues for the development of efficient Grid environments. Several approaches co-exist in today's middleware platforms. The computational (or communication) granularity and the dependences between the computations also have a great influence on the software choices. Two possible approaches are identified below.

One approach provides the user with a uniform view of resources. This is the case of GLOBUS [1] which provides transparent MPI communications (with MPICH-G2) between distant nodes but does not manage load balancing issues between these nodes. It is the user's task to develop a code that will take the heterogeneity of the target architecture into account. The classical batch processing paradigm can also be used on the Grid with projects like Condor-G [2] or Sun GridEngine [3]. Finally, peer-to-peer [99] or Global computing [109] can be used for fine grain and loosely coupled applications.

Another approach provides a semi-transparent access to computing servers by submitting jobs to dedicated servers. This model is known as the Application Service Provider (ASP) model where providers offer, not necessarily for free, computing resources (hardware and software) to clients in the same way as Internet providers offer network resources to clients. The programming granularity of this model is rather coarse. One of the advantages of this approach is that end users do not need to be experts in parallel programming to benefit from high performance parallel programs and computers. This model is closely related to the classical Remote

[1] http://www.globus.org/
[2] http://www.cs.wisc.edu/condor/condorg/
[3] http://wwws.sun.com/software/gridware/

Procedure Call (RPC) paradigm. On a Grid platform, the RPC (or GridRPC [111], [112]) offers an easy access to available resources to a Web browser, a Problem Solving Environment, or a simple client program written in C, Fortran, or Java. It also provides more transparency by hiding the search and allocation of computing resources. We favor this second approach.

In a Grid context, the second approach requires the implementation of middleware environments to facilitate the client access to remote resources. In the ASP approach, a common way for clients to ask for resources to solve their problem is to submit a request to the middleware. The middleware finds the most appropriate server that will solve the problem on behalf of the client using a specific software. Several environments, usually called Network Enabled Servers (NES), have developed such a paradigm: NetSolve [101], Ninf [113], NEOS [108], OmniRPC [115], and more recently DIET developed in the GRAAL project (see Section 5.1). A common feature of these environments is that they are built on top of five components: clients, servers, databases, monitors, and schedulers. Clients solve computational requests on servers found by the NES. The NES schedules the requests on the different servers using performance information obtained by monitors and stored in a database.

Two axis of generalization of this issue can be pursued. The first one is with respect to the targeted infrastructure. More volatile and insecure contexts such as desktop computing appear also important to be considered. The second axis consists in taking into account other forms of interactions than RPC. A general conceptual model for dealing with it is represented by software component models.

To achieve our goals, we need to address issues related to several well-known research domains. In particular, we focus on:

- middleware and application platforms as a base to implement the necessary “glue” to broker clients requests, to find the best server available, and then to submit the problem and its data,
- online and offline scheduling of requests,
- link with data management,
- distributed algorithms to manage the requests and the dynamic behavior of the platform,
- programming models to offer an adequate level of functionality while hiding as many resource related details as possible.

4. Application Domains

4.1. Applications of Sparse Direct Solvers

In the context of our activity on sparse direct (multifrontal) solvers in distributed environments, we develop, distribute, maintain and support competitive software. Our methods have a wide range of applications, and they are at the heart of many numerical methods in simulation: whether a model uses finite elements or finite differences, or requires the optimization of a complex linear or nonlinear function, one almost always ends up solving a linear system of equations involving sparse matrices. There are therefore a number of application fields, among which we list some cited by the users of our sparse direct solver MUMPS (see Section 5.2): structural mechanical engineering (e.g., stress analysis, structural optimization, car bodies, ships, crankshaft segment, offshore platforms, computer assisted design, computer assisted engineering, rigidity of sphere packings); heat transfer analysis; thermomechanics in casting simulation; fracture mechanics; biomechanics; medical image processing; tomography; plasma physics (e.g., Maxwell’s equations), critical physical phenomena, geophysics (e.g., seismic wave propagation, earthquake related problems); ad-hoc networking modeling (e.g., Markovian processes); modeling of the magnetic field inside machines; econometric models; soil-structure interaction problems; oil reservoir simulation; computational fluid dynamics (e.g., Navier-Stokes, ocean/atmospheric modeling with mixed finite elements methods, fluvial hydrodynamics, viscoelastic flows); electromagnetics; magneto-hydro-dynamics; modeling the structure of the optic nerve head and of cancellous bone; modeling of the heart valve; modeling and simulation of crystal growth processes; chemistry (e.g., chemical process modeling); vibro-acoustics; aero-acoustics; aero-elasticity; optical fiber modal analysis; blast furnace modeling;

glaciology (e.g., modeling of ice flow); optimization; optimal control theory; astrophysics (e.g., supernova, thermonuclear reaction networks, neutron diffusion equation, quantum chaos, quantum transport); research on domain decomposition (e.g., MUMPS is used on subdomains in an iterative solver framework); and circuit simulations.

4.2. Molecular Dynamics

LAMMPS is a classical molecular dynamics (MD) code created for simulating molecular and atomic systems such as proteins in solution, liquid-crystals, polymers, or zeolites. It was designed for distributed-memory parallel computers and runs on any parallel platform that supports the MPI message-passing library or on single-processor workstations. LAMMPS is mainly written in F90.

LAMMPS was originally developed as part of a 5-way DoE-sponsored CRADA collaboration between 3 industrial partners (Cray Research, Bristol-Myers Squibb, and Dupont) and 2 DoE laboratories (Sandia and Livermore). The code is freely available under the terms of a simple license agreement that allows you to use it for your own purposes, but not to distribute it further.

The integration of LAMMPS into our Problem Solving Environment DIET is in progress. Discussions are still taking place in order to make the LAMMPS service available through a web portal, on at least one cluster managed by the Sun Grid Engine batch scheduler.

4.3. Biochemistry

Current progress in different areas of chemistry such as organic chemistry, physical chemistry or biochemistry allows the construction of complex molecular assemblies with predetermined properties. In all these fields, theoretical chemistry plays a major role by helping to build various models which can greatly differ in terms of theoretical and computational complexity, and which allow the understanding and the prediction of chemical properties.

Among the various theoretical approaches available, quantum chemistry is at a central position as all modern chemistry relies on it. This scientific domain is quite complex and involves heavy computations. In order to fully apprehend a model, it is necessary to explore the whole potential energy surface described by the independent variation of all its degrees of freedom. This involves the computation of many points on this surface.

Our project is to couple DIET with a relational database in order to explore the potential energy surface of molecular systems using quantum chemistry: all molecular configurations to compute are stored in a database, the latter is queried, and all configurations that have not been computed yet are passed through DIET to computer servers which run quantum calculations, all results are then sent back to the database through DIET. At the end, the database will store a whole potential energy surface which can then be analyzed using proper quantum chemical analysis tools.

4.4. Bioinformatics

Genomics acquiring programs, such as full genomes sequencing projects, are producing larger and larger amounts of data. The analysis of these raw biological data require very large computing resources. In some cases, due to the lack of sufficient computing and storage resources, skilled staff or technical abilities, laboratories cannot afford such huge analyses. Grid computing may be a viable solution to the needs of the genomics research field: it can provide scientists with a transparent access to large computational and data management resources.

In this application domain, we are currently addressing two different problems. In the first one, we tackle the problem of clustering the sequences contained in international databanks into domain protein families. Our aim is to ensure, through the use of grids, the capacity of timely and automatically building of databases (such as ProDom) when such databases are built from exponentially-fast growing protein databases.

In the second problem, we consider protein functional sites. Functional sites and signatures of proteins are very useful for analyzing raw biological data or for correlating different kinds of existing biological data. These methods are applied, for example, to the identification and characterization of the potential functions of new sequenced proteins. The sites and signatures of proteins can be expressed by using the syntax defined by the PROSITE databank, and written as a "protein regular expression". Searching one such site in a sequence can be done with the criterion of the identity between the searched and the found patterns. Most of the time, this kind of analysis is quite fast. However, in order to identify non perfectly matching but biologically relevant sites, the user can accept a certain level of error between the searched and the matching patterns. Such an analysis can be very resource consuming.

4.5. Cosmological Simulations

Ramses [4] is a typical computational intensive application used by astrophysicists to study the formation of galaxies. *Ramses* is used, among other things, to simulate the evolution of a collisionless, self-gravitating fluid called "dark matter" through cosmic time. Individual trajectories of macro-particles are integrated using a state-of-the-art "N body solver", coupled to a finite volume Euler solver, based on the Adaptive Mesh Refinement technique. The computational space is decomposed among the available processors using a *mesh partitioning* strategy based on the Peano-Hilbert cell ordering.

Cosmological simulations are usually divided into two main categories. Large scale periodic boxes requiring massively parallel computers are performed on a very long elapsed time (usually several months). The second category stands for much faster small scale "zoom simulations". One of the particularity of the HORIZON project is that it allows the re-simulation of some areas of interest for astronomers.

We designed a Grid version of *Ramses* through the DIET middleware. From Grid'5000 experiments we proved that DIET is capable of handling long cosmological parallel simulations: mapping them on parallel resources of a Grid, executing and processing communication transfers. The overhead induced by the use of DIET is negligible compared to the execution time of the services. Thus DIET permits to explore new research axes in cosmological simulations (on various low resolutions initial conditions), with transparent access to the services and the data.

4.6. Ocean-Atmosphere Simulations

Climatologists have recourse to numerical simulation and particularly coupled models in several occasions: for example, to estimate natural variability (thousand of simulated years), for seasonal forecasting (only a few simulated months) or to study global warming characteristics (some simulated decades).

To take advantage of the Grid'5000 platform, we choose to launch parallel simulations (ensemble) on several nodes, approximatively 10 or more, according to the load of the platform. Scenario simulations, that simulate from present climate to the next century, require huge computing power. Indeed, each simulation will differ from each other in physical parameterization of atmospheric model. Comparing them, we expect to better estimate global warming prediction sensibility in order to model parameterization.

Practically, a 150 year long scenario combines 1800 simulations of one month each, launched one after the other. This partitioning eases workflow and implements checkpointing because the ending state of the simulation of one month is used as the initial state of the next month.

Our goal regarding the climate forecasting application is to thoroughly analyze it in order to model its needs in terms of execution model, data access pattern, and computing needs. Once a proper model of the application has been derived, appropriate scheduling heuristics can be proposed, tested, and compared. We plan to extend this work to provide generic scheduling schemes for applications with similar dependence graphs.

[4] http://www.projet-horizon.fr/

4.7. Décrypthon

The Décrypthon project is built over a collaboration between CNRS, AFM (*Association Française contre les Myopathies*), and IBM. Its goal is to make computational and storage resources available to bioinformatic research teams in France. These resources, connected as a Grid through the Renater network, are installed in six universities and schools in France (Bordeaux, Jussieu, Lille, Lyon, Orsay, and Rouen). The Décrypthon project offers means necessary to use the Grid through financing of research teams and postdoc, and assistance on computer science problems (such as modeling, application development, and data management). The GRAAL research team is involved in this project as an expert for application gridification. The Grid middleware used at the beginning of the project was GridMP from United Devices. In 2007, DIET was chosen to be the Grid middleware of the Décrypthon Grid. It ensures the load-balancing of jobs over the six computation centers through the Renater network. This transfer of our middleware, first built for large scale experimentations of scheduling heuristics, in a production Grid is a real victory for our research team.

In 2009, we have made updates of the existing applications on the Decrypthon grid (MS2PH and Docking/MaxDO). We also added features to the DIET WebBoard (web interface for managing the Decrypthon Grid through DIET): possibility to use a postgreSQL database instead of a MySQL one, support for parallel jobs, a new search page, functions to manage storage space, ... We are now porting a new BLAST application and an OMSSA application on the Decrypthon grid.

The MaxDO "Help cure muscular dystrophy, phase 2" was ported on the World Community Grid. To determine the size of the work-units sent to the World Community Grid users we ran benchmarks on Grid'5000. Finally on May 14th 2009 the project was launched and it is running since then. On December 16th 2009 a total of 9,507,842 work-unit results had been sent back by the World Community Grid volunteers, this is 13,061,972,568 positions out of 137,652,178,995 (9.49% of the project, each work-unit contains hundreds of "positions" for two proteins: the result is an energy value for this configuration). We are also sorting the result files, reducing their size, and making statistics for the volunteers (cf http://graal.ens-lyon.fr/~nbard/WCGStats/). The estimated end of the project is for the end of 2011.

4.8. Micro-Factories

Micro-factories are automated units designed to produce pieces composed of micro-metric elements. Today's micro-factories are composed of elementary modules or robots able to carry out basic operations. To perform more complex operations, few elementary modules may be grouped in a cell. The realization of one of these cells is still a scientific challenge but several research projects have already got significant results in this domain. These results show very promising functionalities like the ability to configure or reconfigure a cell, by changing a robot tool for instance. However, the set of operations carried out by a cell is still limited. The next generation of micro-factories will put several cells together and make them cooperate to produce complex assembled pieces, as we do for macroscopic productions. In this context, the cell control will evolve to become more cooperative and distributed.

Micro-factories may be modeled in a way that allows to reuse the results obtained in scheduling on heterogeneous platforms as Grids, in particular the results on steady-state scheduling. We develop scheduling strategies and algorithms adapted to this context and we optimize the deployment of cells based on the micro-product and the production specification. We are currently working on the evaluation and the adaptation of several scheduling algorithms in this context, taking small-to-medium batch of jobs into account.

At the micro-metric scale, the manipulation of the elements cannot be considered the same way as at macro-metric scale because the equilibrium of forces is modified. For instance, the electrostatic force becomes predominant on the gravity. This lead to uncontrolled behaviors and frequently generates faults. We are working on taking these faults into account into scheduling models and evaluating their performance depending on the fault characteristics.

5. Software

5.1. DIET

Participants: Nicolas Bard, Raphaël Bolze, Yves Caniou, Eddy Caron, Ghislain Charrier, Frédéric Desprez [correspondent], Jean-Sébastien Gay, Vincent Pichon.

Huge problems can now be processed over the Internet thanks to Grid Computing Environments like Globus or Legion. Because most of the current applications are numerical, the use of libraries like BLAS, LAPACK, ScaLAPACK, or PETSc is mandatory. The integration of such libraries in high level applications using languages like Fortran or C is far from being easy. Moreover, the computational power and memory needs of such applications may of course not be available on every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid as explained in Section 3.3. The aim of the DIET project (http://graal.ens-lyon.fr/DIET) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a NES environment such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network.

DIET is built upon *Server Daemons*. The scheduler is scattered across a hierarchy of *Local Agents* and *Master Agents*. Network Weather Service (NWS) [117] sensors are placed on each node of the hierarchy to collect resource availabilities.

The different components of our scheduling architecture are the following. A **Client** is an application which uses DIET to solve problems. Many kinds of clients should be able to connect to DIET from a web page, a Problem Solving Environment such as Matlab or Scilab, or a compiled program. A **Master Agent (MA)** receives computation requests from clients. These requests refer to some DIET problems listed on a reference web page. Then the MA collects computational abilities from the servers and chooses the best one. The reference of the chosen server is returned to the client. A client can be connected to an MA by a specific name server or a web page which stores the various MA locations. Several MAs can be deployed on the network to balance the load among them. A **Local Agent (LA)** aims at transmitting requests and information between MAs and servers. The information stored on a LA is the list of requests and, for each of its subtrees, the number of servers that can solve a given problem and information about the data distributed in this subtree. Depending on the underlying network topology, a hierarchy of LAs may be deployed between an MA and the servers. No scheduling decision is made by a LA. A **Server Daemon (SeD)** encapsulates a computational server. For instance it can be located on the entry point of a parallel computer. The information stored on a SeD is a list of the data available on its server (with their distribution and the way to access them), the list of problems that can be solved on it, and all information concerning its load (available memory and resources, etc.). A SeD declares the problems it can solve to its parent LA. A SeD can give performance prediction for a given problem thanks to the CoRI module (Collector of Resource Information) [103]. Master Agents can then be connected over the net (Multi-MA version of DIET), either statically or dynamically.

Moreover applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers* [103]. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

Tools have recently been developed to deploy the platform (GoDIET), to monitor its execution (LogService), and to visualize its behavior using Gantt graphs and statistics (VizDIET).

Seen from the user/developer point of view, the compiling and installation process of DIET should remain simple and robust. But DIET has to support this process for an increasing number of platforms (Hardware architecture, Operating System, C/C++ compilers). Additionally DIET also supports many functional extensions

(sometimes concurrent) and many such extensions require the usage of one or a few external libraries. Thus the compilation and installation functionalities of DIET must handle a great number and variety of possible specific configurations. Up to the previous versions, DIET's privileged tool for such a task were the so-called GNU-autotools. DIET's autotools configuration files evolved to become fairly complicated and hard to maintain. Another important task for the packaging DIET is to assess that DIET can be properly compiled and installed at least for the most mainstream platforms and for a decent majority of all extension combinations. This quality assertion process should be realized with at least the frequency of the release. But, as clearly stated by the agile software development framework, the risk can be greatly reduced by developing software in short time-boxes (as short as a single cvs commit). For the above reasons, it was thus decided to move away from the GNU-autotools to cmake (refer http://www.cmake.org). Cmake offers a much simpler syntax for its configuration files (sometimes at the cost of semantics, but cmake remains an effective trade-off). Additionally, cmake integrates a scriptable regression test tool whose reports can be centralized on a so called dashboard server. The dashboard offers a synthetic view (see http://graal.ens-lyon.fr/DIET/dietdashboard.html) of the current state of DIET's code. This quality evaluation is partial (compilation and linking errors and warnings) but is automatically and constantly offered to the developers. Although the very nature of DIET makes it difficult to carry distributed regression tests, we still hope that the adoption of cmake will significantly improve DIET's robustness and general quality.

DIET has been validated on several applications. Some of them have been described in Sections 4.2 through 4.7.

5.1.1. Workflow support

Workflow-based applications are scientific, data intensive applications that consist of a set of tasks that need to be executed in a certain partial order. These applications are an important class of Grid applications and are used in various scientific domains like astronomy or bioinformatics.

We have developed a workflow engine in DIET to manage such applications and propose to the end-user and the developer a simple way either to use provided scheduling algorithms or to develop their own scheduling algorithm.

There are many Grid workflow frameworks that have been developed, but DIET is the first GridRPC middleware that provides an API for workflow execution. Moreover, existing tools have limited scheduling capabilities. One of our objectives is to provide an open system which provides several scheduling algorithms, but also that allows users to plug and use their own specific schedulers.

In our implementation, workflows are described using the XML language. Since no standard exists for scientific workflows, we have proposed our formalism. The DIET agent hierarchy has been extended with a new special agent, the *MA_DAG*. To be flexible we can execute workflows even if this special agent is not present in the platform. The use of the *MA_DAG* centralizes the scheduling decisions and thus can provide a better scheduling when the platform is shared by multiple clients. On the other hand, if the client bypasses the *MA_DAG*, a new scheduling algorithm can be used without affecting the DIET platform. The current implementation of DIET provides several schedulers (Round Robin, HEFT, random, Fairness on finish Time, etc.).

The DIET workflow runtime also includes a rescheduling mechanism. Most workflow scheduling algorithms are based on performance predictions that are not always accurate (erroneous prediction tool or resource load wrongly estimated). The rescheduling mechanism can trigger the application rescheduling when some conditions specified by the client are filled.

We also continued our work on schedulers for DIET workflow engine concerning multi-workflows based applications, and graphical tools for workflows within the DIET DashBoard project. Within the Gwendia project, we worked on the implementation of the language defined in the project and around the Cardiac application. Experiments were done over the Grid'5000 platform.

5.1.2. Batch and parallel job management

Generally, the use of a parallel computing resource is done through a batch reservation system. Users wishing to submit parallel tasks have to write *scripts* which notably describe the number of required nodes and the walltime of the reservation. Once submitted, a script is processed by the batch scheduling algorithm: the user is answered the starting time of its job, and the batch system records the dedicated nodes (*the mapping*) allocated to the job.

In the Grid context, there is consequently a two-level scheduling: one at the batch level and the other one at the Grid middleware level. In order to efficiently exploit the resource (according to some metrics), the Grid middleware should map the computing tasks according to the local scheduler policy. This also supposes that the middleware integrates some mechanisms to submit to parallel resources, and that, during the submission, it provides information like the number of demanded resources, the job deadline, etc.

DIET servers are able to transparently submit tasks to parallel resources, via a batch system or not. For the moment, DIET servers can submit to the version 1.6 and 2.X of OAR, OpenPBS and Loadleveler reservation systems, the latter being used in the Décrypthon project. The implementation of the integration of SGE is in progress. Functions to access batch system information have also been implemented in order to use them both as scheduling metric and to tune parallel and moldable tasks.

5.1.3. DIET Data Management

DAGDA, designed during the PhD of Gaël Le Mahec, is a new data manager for the DIET middleware which allows data explicit or implicit replications and advanced data management on the grid. It was designed to be backward compatible with previously developed applications for DIET which benefit transparently of the data replications. It allows explicit or implicit data replications, file sharing between the nodes which can access to the same disk partition, the choice of a data replacement algorithm, and a high level configuration about the memory and disk space DIET should use for the data storage and transfers.

To transfer a data, DAGDA uses the pull model instead of the push model used by DTM. The data are not sent into the profile from the source to the destination, but they are downloaded by the destination from the source. DAGDA also chooses the best source for a given data.

DAGDA has also been used for the validation of our join replication and scheduling algorithms over DIET.

5.1.4. GridRPC Data Management API

Data Management is a challenging issue inside the OGF GridRPC standard, for performance reasons. Indeed some temporarily data do not need to be transfered once computed and can reside on servers for example. We can also imagine that data can be directly transferred from one server to another one, without being transferred to the client in accordance to the GridRPC paradigm behavior.

We have consequently worked on a Data Management API which has been presented to all OGF sessions since OGF'21. Since december 2009, the proposal is available for public comment and may be reached at: http://www.ogf.org/gf/docs/?public_comment under the title "Proposal for a Data Management API within the GridRPC. Y. Caniou and others, via GRIDRPC-WG".

5.1.5. DIET Dashboard

When the purpose is to monitor a Grid, or deploy a Grid middleware on it, several tasks are involved in the process. Managing the resources of a Grid: allocating resources, deploying nodes with defined operating systems, etc. Monitoring the Grid: getting the status of the clusters (number of available nodes in each state, number and main properties of each job, Gantt chart of the jobs history), the status of the jobs (number, status, owner, walltime, scheduled start, ganglia information of the nodes) present in the platform, etc. Managing Grid middleware in Grid environment: designing hierarchies (manually or automatically by matching resources on patterns), deploying them directly or through workflows of applications, etc.

The DIET Dashboard provides tools trying to answer these needs with an environment dedicated to the GridRPC middleware DIET and it consists of a set of graphical tools that can be used separately or together.

These tools can be divided in three categories:

DIET tools including tools to design and deploy DIET applications. The DIET Designer allows users to graphically design a DIET hierarchy. The DIET Mapping tool allows users to map the allocated Grid'5000 resources to a DIET application. The mapping is done in an interactive way by selecting the site then DIET agents or SeDs. And the DIET Deploy tool is a graphical interface to GoDIET intended for the deployment of DIET hierarchies.

Workflow tools including workflow designer and workflow log service. **The Workflow designer** is dedicated to workflow applications written in DIET. It gives users an easy way to design and execute workflows. The user can compose the available services and link them by drag-and-drop or load a workflow description file in order to reuse it. Finally it can be directly executed online. **The Workflow LogService** can be used to monitor workflows execution by displaying the DAG nodes of each workflow and their states.

Grid tools (aka GRUDU). These tools are used to manage, monitor, and access user Grid resources. **Displaying the status of the platform**: this feature provides information about clusters, nodes and jobs. **Resource allocation**: this feature provides an easy way to allocate resources by selecting from a Grid'5000 map the number of required nodes and defining time. The allocated resources can be stored and used with DIET mapping tool. **Resource monitoring** through the use of the Ganglia plugin that provides low-level information on every machines of a site (instantaneous data) or on every machines of a job (history of the metrics). **Deployment management** with a GUI for KaDeploy simplifying its use. **A terminal emulator** for remote connections to Grid'5000 machines and a File transfer manager to send/receive files to/from Grid'5000 frontends.

As the Grid tools can be a powerful help for the Grid'5000 users, these have been extracted to create GRUDU (Grid'5000 Reservation Utility for Deployment Usage) which aims at simplifying the access and the management of Grid'5000.

5.1.6. Middleware Interoperability

For the requirements of the GridTLSE project, DIET has been extended with a protocol interoperability with the ITBL middleware which manages Japanese computing resources in the JAEA (Japan Atomic Energy Agency). A demo has been presented in the INRIA booth at SuperComputing'08 and SuperComputing'09.

5.1.7. DIET as a Cloud System

A new extension of DIET was designed to deal with Cloud platforms such as Amazon EC2. We proposed the use of the DIET Grid middleware on top of the EUCALYPTUS Cloud system to demonstrate general purpose computing using Cloud platforms. DIET is now compliant with the Amazon EC2 API. These recent developments validate the use of a Cloud system as a raw computational on-demand resource for a Grid middleware such as DIET.

5.2. MUMPS

Participants: Maurice Brémond, Indranil Chowdhury, Guillaume Joslin, Jean-Yves L'Excellent [correspondent], Bora Uçar.

MUMPS (for *MUltifrontal Massively Parallel Solver*, see http://graal.ens-lyon.fr/MUMPS) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, mainly in collaboration with ENSEEIHT-IRIT (Toulouse, France), lots of developments have been done, to enhance the software with more functionalities and integrate recent research work. Recent developments also involve the former INRIA project ScAlAppliX since the recruitment of Abdou Guermouche as an assistant professor at *LaBRI*, while CERFACS contributes to some research work.

MUMPS implements a direct method, the multifrontal method, and is a parallel code for distributed memory computers; it is unique by the performance obtained and the number of functionalities available, among which we can cite:

- various types of systems: symmetric positive definite, general symmetric, or unsymmetric,
- several matrix input formats: assembled or expressed as a sum of elemental matrices, centralized on one processor or pre-distributed on the processors,
- detection of null pivots,
- preprocessing and scaling for symmetric and unsymmetric matrices,
- partial factorization and Schur complement matrix,
- dense or sparse right-hand sides, centralized or distributed solution,
- real or complex arithmetic, single or double precision,
- partial threshold pivoting,
- fully asynchronous approach with overlap of computation and communication,
- distributed dynamic scheduling of the computational tasks to allow for a good load balance in the presence of unexpected dynamic pivoting or in multi-user environments.

MUMPS is currently used by more than 1000 academic and industrial users, from a wide range of application fields (see Section 4.1). Notice that the MUMPS users include:

- students and academic users from all over the world;
- various developers of finite element software;
- companies such as Boeing, EADS, EDF, Free Field Technologies, or Samtech.

The latest release is MUMPS 4.9.2, available since November 2009 (see http://graal.ens-lyon.fr/MUMPS/). The most recent features available are: a parallel analysis phase based on the parallel graph partitioning tools pt-scotch or parmetis, the use of 64-bit integers to address large memories, improved performance and memory usage.

5.3. ULCMi

Participants: Julien Bigot, Hinde Bouziane, Christian Pérez [correspondent], Vincent Pichon.

ULCMi is an implementation of the ULCM component model defined in the ANR COSINUS LEGO project. It aims at increasing component model abstraction level for high performance computing by combining component, workflow, data sharing and skeletton concepts.

ULCMi embeds an ULCM interpreter and the adequate runtime systems. It currently supports primitive components written in Java, C++, and also OMG CORBA component. With respect to deployment, Java and C++ components are deployed locally and supports multithreading, while CCM components can be deployed remotely thanks to the use of ADAGE.

5.4. BitDew

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, Wang Yu.

BITDEW is an open source middleware implementing a set of distributed services for large scale data management on Desktop Grids and Clouds. BITDEW relies on 5 abstractions to manage the data : i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of machine crash, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp or bittorrent). Programmers define for every data these simple criteria, and let the BITDEW runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

The current status of the software is the following : BITDEW is open source under the GPLv3 or Cecill licence at the user's choice, 10 releases where produced in the last 2 years and it has been downloaded approximativelly 2500 times on the INRIA forge. Known users are Université Paris-XI, Université Paris-XIII, University of Florida, Cardiff University and University of Sfax.

5.5. XtremWeb

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, Wang Yu.

XTREMWEB is an open source software for Desktop Grid computing, jointly developed by INRIA and IN2P3.

XTREMWEB allows to build lightweight Desktop Grid by gathering the unused resources of Desktop Computers (CPU, storage, network). Its primary features permit multi-users, multi-applications and cross-domains deployments. XTREMWEB turns a set of volatile resources spread over LAN or Internet into a runtime environment executing high througput applications.

XTREMWEB is highly programmable and customizable a wide range of applications (bag-of tasks, master/worker), of computing requirements (data/CPU/network-intensive) and computing infrastructures (clusters, Desktop PCs, multi-Lan) in a manageable, scalable and secure fasion. Known users: LIFL, LIP, LIG, LRI (CS), LAL (physics Orsay), IBBMC (biology), Université Paris-XIII, Université de Guadeloupe, IFP (petroleum), EADS, CEA, University of Wisconsin Madison, University of Tsukuba (Japan), AIST (Australia), UCSD (USA), Université de Tunis, AlmerGrid (NL), Fundecyt (Spain), Hobai (China), HUST (China).

There exists two branches of XTREMWEB: XtremWeb-HEP is a production version developed by IN2P3. It features many security improvements such as X509 support which allows its usage within the EGEE context. Xtremweb-CH is a research version developed by HES-SO, Geneva, which aims at building an effective Peer-To-Peer system for CPU time consuming applications.

XTREMWEB has been supported by national grants (ACI CGP2P) and by major European grants around Grid and Desktop Grid such as FP6 CoreGrid: European Network of Excellence, FP6 Grid4all, and more recently FP7 EDGeS : Enabling Desktop Grid for E-Science.

6. New Results

6.1. Scheduling Strategies and Algorithm Design for Heterogeneous Platforms

Participants: Anne Benoît, Leila Ben Saad, Sékou Diakité, Alexandru Dobrila, Fanny Dufossé, Matthieu Gallet, Mathias Jacquelin, Loris Marchal, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Paul Renaud-Goud, Clément Rezvoy, Yves Robert, Bernard Tourancheau, Frédéric Vivien.

6.1.1. Mapping simple workflow graphs

Mapping workflow applications onto parallel platforms is a challenging problem that becomes even more difficult when platforms are heterogeneous—nowadays a standard assumption. A high-level approach to parallel programming not only eases the application developer's task, but it also provides additional information which can help realize an efficient mapping of the application. We focused on simple application graphs such as linear chains and fork patterns. Workflow applications are executed in a pipeline manner: a large set of data needs to be processed by all the different tasks of the application graph, thus inducing parallelism between the processing of different data sets. For such applications, several antagonist criteria should be optimized, such as throughput, latency, failure probability and energy minimization.

We have considered the mapping of workflow applications onto different types of platforms: *fully homogeneous* platforms with identical processors and interconnection links; *communication homogeneous* platforms, with identical links but processors of different speeds; and finally, *fully heterogeneous* platforms.

Once again, this year we have focused mainly on pipeline graphs, and considered platforms in which processors are subject to failure during the execution of the application. We have added a new optimization objective, namely, the energy minimization, and we have also addressed more sophisticated settings by considering several concurrent applications. On the theoretical side, we have established the complexity of many optimization problems involving energy and several concurrent applications. On the experimental side, we have designed several heuristics which aim at efficiently mapping concurrent applications on a heterogeneous platform (for an energy criterion), given some constraints on the throughput and latency of the application.

Also, in a joint work with Kunal Agrawal (at MIT during the course of the study), we have thoroughly investigated the complexity of the *scheduling* problem: given a mapping, it turns out that it is difficult to orchestrate communication and computation operations, i.e., to decide at which time-step each operation should begin (and end). We demonstrated that some instances of this problem are NP-hard, and we provided some approximation algorithms.

Finally, in collaboration with Oliver Sinnen, from University of Auckland (New Zealand), we have investigated the bi-criteria problem of both throughput and reliability optimization, when processors are subject to failures. The mechanism of replication, which refers to the mapping of an application stage onto more than one processor, can be used to increase throughput but also to increase reliability. Finding the right replication trade-off plays a pivotal role for this bi-criteria optimization problem. Our formal model includes heterogeneous processors, both in terms of execution speed as well as in terms of reliability. We have studied the complexity of the various subproblems and shown how a solution can be obtained for the polynomial cases. For the general NP-hard problem, we have proposed heuristic algorithms, which were experimentally evaluated. We have also proposed the design of an exact algorithm based on A* state space search which allows us to evaluate the performance of our heuristics for small problem instances.

6.1.2. Resource allocation strategies for in-network stream processing

We pursued the work on the operator mapping problem for in-network stream processing applications, initiated last year. In-network stream processing consists in applying a tree of operators in steady-state to multiple data objects that are continually updated at various locations on a network. Examples of in-network stream processing include the processing of data in a sensor network, or of continuous queries on distributed relational databases. Last year, we focused on a "constructive" scenario, i.e., a scenario in which one builds a platform dedicated to the application by purchasing processing servers with various costs and capabilities. The objective was to minimize the cost of the platform while ensuring that the application achieves a minimum steady-state throughput. This year, we considered a more general non-constructive scenario and investigated the problem in which several applications are using the platform concurrently. In particular, we demonstrated the importance of node reuse in such a context: if we can reuse some results from one application to another, we decrease the load on the processors while adding some more communication. Several sophisticated heuristics have been designed and evaluated.

6.1.3. Scheduling small to medium batches of identical jobs

Steady-state scheduling is optimal for an infinite number of jobs. It defines a schedule for a subset of jobs which are performed into a period. The global schedule is obtained by infinitely repeating this period. In the case of a finite number of jobs, this scheduling technique can however be used if the number of computed jobs is large. Three phases are distinguished in the schedule: an initialization phase which computes the tasks needed to enter the steady state, an optimal phase composed of several full periods and a termination phase which finishes the tasks remaining after the last period. With a finite number of jobs we must consider a different objective function, the makespan, instead of the throughput used in the steady-state case. We know that the steady-state phase of the schedule is optimal, thus we are interested in optimizing the initial and final phases.

We have worked on the improvement of the steady-state technique for a finite number of jobs. The main idea is to improve the scheduling of the sub-optimal phases: initialization and termination. By optimizing these two phases we reduce their weight in the global schedule and thus improve its performance. In the

original algorithm the period is computed by a linear program. As a result, the period's length can be quite large, resulting in a lot of temporary job instances. Each of these temporary job instances must be prepared in the initialization phase, and finished in the termination phase. We have two directions of optimization: (i) limiting the period length using sub-optimal but much simpler solutions, and (ii) better organizing the period to reduce the number of inter-period dependencies. Both propositions have been studied and implemented in the SimGrid toolkit. We have demonstrated the usefulness of both approaches (and their combination) to obtain a steady-state schedule more suited when the number of jobs to process is a few hundreds.

6.1.4. Static strategies for worksharing with unrecoverable interruptions

In this work, one has a large workload that is "divisible" and one has access to a number of remote computers that can assist in computing the workload. The problem is that the remote computers are subject to interruptions of known likelihood that kill all work in progress. One wishes to orchestrate sharing the workload with the remote computers in a way that maximizes the expected amount of work completed. In a previous work, we studied strategies for achieving this goal, by balancing the desire to checkpoint often, in order to decrease the amount of vulnerable work at any point, vs. the desire to avoid the context-switching required to checkpoint. This study was done when interruptions are following a linear model, and when the remote computers have the same characteristics.

We first extended that initial study by showing that the heuristics we designed could be straightforwardly extended to deal with any failure model. We validated this extension by simulating these heuristics using actual traces.

We also extended our initial study by considering heterogeneous platforms where the remote computers can be connected with different bandwidths, have different computing speeds, or be subject to different failure laws, as long as these laws are all linear. When at least two of the three computers' characteristics are homogeneous, we proposed closed-form formulas or recurrences to derive the optimal solution. This was done under the hypothesis that the whole divisible load is distributed to computers, and that the work is distributed in a single round. We exposed the complexity of the general case.

6.1.5. Scheduling identical jobs with unreliable tasks

Depending on the context, the fault tolerance model may differ. We have studied the case where the fault probability depends on the tasks instead of on the execution resources. The practical use case is a micro-factory where operations are performed on microscopic components. Due to the size of the components, some operations are not as well controlled as the others and thus the complexity of a task has impacts on the task's reliability. In this context, we consider the schedule of a set of identical jobs composed of either linear chains or trees of tasks. Several objectives are studied depending on the available resources, in particular maximizing the throughput (number of components output per time unit), and minimizing the makespan (total time needed to output the required number of components). The resources in use are heterogeneous and general purpose but must be configured to execute a determined task type. For this reason, finding a good schedule turns into an assignment problem. The most simple instances of this problem can be solved in polynomial time whereas the other cases are NP-complete; for those cases, we designed polynomial heuristics to solve the problem.

This year, we focused on the case in which the failure probability may depend both on tasks and on execution resources, and developed more heuristics. Also, we were able to derive a linear programming formulation of the problem and thus to assess the absolute performance of our heuristics.

6.1.6. Resource allocation using virtual clusters

We proposed a novel job scheduling approach for sharing a homogeneous cluster computing platform among competing jobs. Its key feature is the use of virtual machine technology for sharing resources in a precise and controlled manner. We justified our approach and proposed several job scheduling algorithms. We presented results obtained in simulations for synthetic and real-world High Performance Computing (HPC) workloads, in which we compared our proposed algorithms with standard batch scheduling algorithms. We found that our approach provides drastic performance improvements over batch scheduling. In particular, we identified a few promising algorithms that perform well across most experimental scenarios. Our results demonstrate

that virtualization technology coupled with lightweight scheduling strategies affords dramatic improvements in performance for HPC workloads. The key advantage of our approach over current cluster sharing solutions is that it increases cluster utilization while optimizing a user-centric metric that captures both notions of performance and fairness, the maximum stretch. A key feature of our approach is that we do not assume any knowledge on the job running times and, thus, work in a non-clairvoyant setting.

6.1.7. Steady-state scheduling of dynamic bag-of-tasks applications

This work focused on sets of independent tasks ("bag-of-tasks" applications) and a simple master-worker platform. In this context, a main processor initially owns all the tasks and distributes them to a pool of secondary processors, or workers, which process the tasks. The aim is then to maximize the average number of tasks processed by the platform per time unit. In this work, all tasks of a bag-of-tasks application do not have the same computation and communication sizes, but these sizes are defined by the distribution of a random variables. This enables to model the inevitable variations between the multiple tasks of an application. The distribution is not supposed to be known, but to be empirically discovered when considered an initial subset of the tasks submitted to the system (say, the first 100 tasks). We presented a method to obtain an ε-approximation of an optimal schedule in case of a continuous flow of instances, as well as several heuristics. The quality of the different solutions were assessed through simulations. The proposed methods are compared to standard algorithms like a Round-Robin distribution or an On-Demand method. The simulations showed that a little knowledge about applications is sufficient to really improve scheduling results, and that steady-state static methods have significantly better performance when communication and computations costs are of the same magnitude. For the cases where either the communications or the computations significantly dominate, the on-demand dynamic method is shown to be asymptotically optimal.

6.1.8. Parallelizing the construction of the ProDom database

ProDom is a protein domain family database automatically built from a comprehensive analysis of all known protein sequences. ProDom development is headed by Daniel Kahn (INRIA project-team BAMBOO, formerly HELIX). With the protein sequence databases increasing in size at an exponential pace, the parallelization of MkDom2, the algorithm used to build ProDom, has become mandatory (the original sequential version of MkDom2 took 15 months to build the 2006 version of ProDom and would have required at least twice that time to build the 2007 version).

The parallelization of MkDom2 is not a trivial task. The sequential MkDom2 algorithm is an iterative process, and parallelizing it involves forecasting which of these iterations can be run in parallel and detecting and handling dependency breaks when they arise. We have moved forward to be able to efficiently handle larger databases. Such databases are prone to exhibit far larger variations in the processing time of query-sequences than was previously imagined. The collaboration with BAMBOO on ProDom continues today both on the computational aspects of the constructing of ProDom on distributed platforms, as well as on the biological aspects of evaluating the quality of the domains families defined by MkDom2, as well as the qualitative enhancement of ProDom. This past year was devoted to improve the new parallel MPI_MkDom2 algorithm and code, for it to be usable in a production setting. Among other improvements, the code was ported to run on the BlueGene/P machine from IDRIS.

6.1.9. Steady-state scheduling on the CELL processor

In this work, we have considered the problem of scheduling streaming applications described by complex task graphs on a heterogeneous multicore processor, the STI Cell BE processor. To this goal, we have proposed a theoretical model of the Cell processor. Then, we have used this model to express the problem of maximizing the throughput of a streaming application on this processor. Although the problem is proven NP-complete, we have presented an optimal solution based on mixed linear programming. This allows us to compute the optimal mapping for a number of applications, ranging from a real audio encoder to complex random task graphs. These mappings have been tested on two real platforms embedding Cell processors, and compared to simple heuristic solutions. We have shown that this mappings allows to achieve a good speed-up, whereas the heuristic solutions generally fail to deal with the strong memory and communication constraints of the

CELL processors. We are currently extending this work to cope with the complex architecture of the IBM Bladecenter QS 22, which embeds two CELL processors.

6.1.10. Fair distributed scheduling of bag-of-tasks applications on desktop grids

Desktop Grids have become very popular nowadays, with projects that include hundred of thousands computers. Desktop grid scheduling faces two challenges. First, the platform is volatile, since users may reclaim their computer at any time, which makes centralized schedulers inappropriate. Second, desktop grids are likely to be shared among several users, thus we must be particularly careful to ensure a fair sharing of the resources.

In this work, we have proposed a distributed scheduler for bag-of-tasks applications on desktop grids, which ensures a fair and efficient use of the resources. It aims to provide a similar share of the platform to every application by minimizing their maximum stretch, using completely decentralized algorithms and protocols. This approach has been validated through extensive simulation. We have shown that its performance is close to the best centralized algorithms for fair scheduling, for a limited bandwidth consumption. This work was conducted in collaboration with Javier Celaya, from the University of Saragosse (Spain).

6.2. Providing access to HPC servers on the Grid

Participants: Nicolas Bard, Julien Bigot, Laurent Bobelin, Raphaël Bolze, Yves Caniou, Eddy Caron, Ghislain Charrier, Florent Chuffart, Benjamin Depardon, Frédéric Desprez, Gilles Fedak, Jean-Sébastien Gay, Haiwu He, Benjamin Isnard, Michael Heymann, Cristian Klein, Gaël Le Mahec, David Loureiro, Georges Markomanolis, Adrian Muresan, Hidemoto Nakada, Christian Pérez, Franck Petit, Vincent Pichon, Bing Tang, Daouda Traore, Wang Yu.

6.2.1. Service Discovery in Peer-to-Peer environments

We have published an extended version [23] of a work started in 2007 around the snap-stabilization of the Distributed Lexicographic Placement Table (DLPT) approach, building a prefix-tree based overlay network for an efficient peer-to-peer service discovery system for grids. Our approach is an alternative choice to inject fault-tolerance once replication, which is mainly used in similar systems, has failed. Moreover, replication can be very costly in terms of computing and storage resources and does not ensure the recovery of the system after arbitrary failures. Self-stabilization is an efficient approach to design reliable solutions for dynamic systems. It ensures a system to converge to its intended behavior, regardless of its initial state, in a finite time. A snap-stabilizing algorithm guarantees that it always behaves according to its specification, once the protocol is launched. We have provided the first snap-stabilizing protocol for tree construction. The proposed algorithm transforms an arbitrary labelled tree into a consistent prefix tree, in average, in O($h + h'$) rounds, where h and h' are the initial and final heights of the tree, respectively. In the worst case, the algorithm requires an O(n) extra space on each node, O(n) rounds and O(n^2) actions. New simulations have been conducted, allowing to state that the worst cases are far from being reached and confirm the average complexities.

We have published a book chapter [85] of a more popularizing view of the DLPT architecture intended to be spread among the computer science community. The results presented summarize the DLPT approach, from early design to its fault-tolerant mechanisms offering formal guarantees in very dynamic and faulty platforms, via its use and load balancing mechanisms. The presentation of the chapter radically differs from previous technical papers on the same results while giving a global view of the work pursued in this area by the GRAAL team.

6.2.2. Deployment of hierarchical middleware

We consider the placement of the various elements of a hierarchical grid middleware. We consider the case where several services have to be deployed within the hierarchy (the case where only one service has to be made available has already been studied), and study several kinds of models and platforms. Our goal is to have fairness between the throughputs of different services, i.e., the ratio between the requested throughput and the obtained throughput should roughly be the same for each service. We studied two models: the first and simplistic one states that whenever a message is received at a given level, whatever the type of service it refers

to, the message is sent to all the children; the second one forwards a message to a children only if the latter knows about this service. We then derived a closed form solution for the simple model on a homogeneous platform, and a bottom up heuristic for the more general model on both homogeneous platforms and platforms with heterogeneous computations but homogeneous communications. We also derived a genetic algorithm for totally heterogeneous platforms.

6.2.3. Scheduling of independent tasks under cluster availability constraints

We consider the scheduling of independent tasks on a grid of clusters, under the constraint that on each cluster a task can be scheduled only if on a given period of time the cluster load is not higher than a given upper bound. We described the problem using a mixed integer linear program. However, as the problem involves lots of variables and constraints, it is much too complicated to be solved this way. Hence, we designed a set of heuristics to solve the problem.

6.2.4. k-clustering

Mobile ad hoc networks as well as grid platforms are distributed, changing, and error prone environments. Communication costs within such infrastructures can be improved, or at least bounded, by using k-clustering. A k-clustering of a graph is a partition of the nodes into disjoint sets, called clusters, in which every node is at distance at most k from a designated node in its cluster, called the clusterhead. We designed a self-stabilizing asynchronous distributed algorithm for constructing a k-clustering of a connected network of processes with unique IDs and weighted edges. The algorithm is comparison-based, takes O(nk) time, and uses O($log(n) + log(k)$) space per process, where n is the size of the network. Using simulations, we show that even if the complexity theoretical bound can be attained on particular graphs, our algorithm requires much less time to converge on many graphs. This is the first distributed solution to the k-clustering problem on weighted graphs.

6.2.5. Proof of concept of ULCM and recursive applications

In the the ANR LEGO project, we have designed a ULCM, a component model which combines various kinds of composition operators. In addition to the classical provide/use composition operators, ULCM also offers data sharing, master-worker and workflow operators. Hence, ULCM unifies classical component models with workflow based models.

In 2009, our main effort was devoted to the realization of ULCMi, a proof-of-concept implementation of ULCM. ULCMi is based on standard compiler technology (`antlr`) that is used to build a representation of a program. Then, the ULCMi runtime is responsible for creating, connecting and running components. As ULCM supports workflows within composite components, a specific workflow engine is also part of the runtime.

ULCMi currently has four back-ends: a simulator back-end to test the validy of a program, a multithreaded `Java` back-end and a `C++` one for local execution so as to study its application to multicore machines, and a CCM based back-end for distributed execution. The `C++` multithreaded back-end is used in particular within the ANR NUMASIS project. It is interfaced with the thread library `marcel`, developed by the RUNTIME project-team.

In cooperation with EDF R&D, we have started studying the support of recursive algorithms in the context of ULCM. A particularly important use case is represented by adaptive mesh refinement applications which are particularly complex to implement. Preliminary results show that ULCM appears expressive enough. However, some questions remains open such as the simplicity of programming – which could be solved by generic components – and by the smallest level of granularity that can be reached while achieving high performance. Hence, the technology appears adequate for coarse and medium grain but the question remains open for fine grain.

6.2.6. Component models and algorithmic skeletons

In 2009, we have conducted a validation of the STKM model. STKM is a component model combining provide/use composition, workflow and algorithmic skeleton operators. STKM can be seen as going one

step further than ULCM as it aims to study the possibility and the benefit of integrating algorithmic skeleton technology within component model in general, and advanced component models such as ULCM in particular. To verify the promises of the model, we have built a proof-of-concept implementation of STKM on top of SCA, a component model based on web services. The proposed mapping of STKM on top of SCA introduces a set of non-functional concerns needed to manage an STKM assembly; concerns that can be hidden to the end user and that can be used for execution optimizations. Hand-coded experiments show that STKM can lead to both better performance and resource usage than a model only based on workflows or skeletons. Hence, the promises of STKM can be achieved provided that the various elements of the model are correctly used. In the general case, it may require to apply optimization algorithms to applications. This work has been done in cooperation with the University of Pisa (Italy).

6.2.7. Component models and genericity

In order to be easily reusable, most component models require a component to be a binary version of a piece of code. However, with respect to most programming languages, this requirement limits reusability as all the types must be fixed. For example, it is not possible to have a general dispatching component for farm component. Moreover, we would like not only to make generic the data-types of component interfaces, but also the interfaces, as well as component types.

To this end, we have pursued our work on increasing reuse in component models by adding the concept of genericity to component models. In order to support dynamic instantiation and to explore the benefit for meta-programming thanks to explicit specializations as a mean to encode algorithmic skeleton, we opted for a solution *à la* `C++`. Hence, all genericity-related features of the model are handled through a compilation phase. In this work, we restrict ourselves to static compilation. Further work may deal with dynamic compilation.

To leverage existing component models, the selected approach is to derive a generic meta-model from an existing one, and to provide an algorithm to transform generic component applications into non-generic ones. This has been applied to SCA, leading to a generic-SCA model. The model transformation algorithm has been implemented within Eclipse and the whole chain has been validated with an image rendering application based on a generic task farm component.

6.2.8. Deployment of hierarchical applications on grids

In the context of the ANR DISCOGRID project, we had developed a model to ease the programming of hierarchical applications, such as computational electromagnetics, on grids. The DISCOGRID model can be seen as an extension of MPI for a hierarchy of resources as well as the addition of hierarchical data redistributions. Another element of the DISCOGRID project was a multilevel partitioning tool that decomposes an unstructured mesh with respect to the available resources. Currently, this tool is limited to two levels, which are typically represented by a federation of clusters. However, as the tool computes a partition of the mesh on *all* resources, the issue was to select a set of resources amongst the available ones to give to such a tool. Because of Amdahl's law and the network latency and bandwidth, taking all the resources does not lead to minimize the execution time. Therefore, we have developed a performance model of a particular CEM application, based on the primitives of the DISCOGRID model. Then, we designed several resource selection algorithms, some based on specific heuristics and some based on generic heuristics such as random or simulated annealing. A series of experiments based on simulations and on real experiments on Grid'5000 demonstrates the validity of the performance model as well as the good accuracy and quick response time of some resource selection algorithms.

6.2.9. Towards Data Desktop Grid

In this work, we have proposed the BITDEW framework which addresses the issue of how to design a programmable environment for automatic and transparent data management on computational Desktop Grids. We described the BITDEW programming interface, its architecture, and the performance evaluation of its runtime components. BITDEW relies on a specific set of meta-data to drive key data management operations, namely life cycle, distribution, placement, replication and fault-tolerance with a high level of abstraction. The BITDEW runtime environment is a flexible distributed service architecture that integrates modular P2P

components such as DHT's for a distributed data catalog and collaborative transport protocols for data distribution. Through several examples, we describe how application programmers and BITDEW users can exploit BITDEW's features. The performance evaluation demonstrates that the high level of abstraction and transparency is obtained with a reasonable overhead, while offering the benefit of scalability, performance and fault tolerance with little programming cost.

Data-intensive applications form an important class of applications for the e-Science community which require secure and coordinated access to large datasets, wide-area transfers and broad distribution of TeraBytes of data while keeping track of multiple data replicas. In computational genomics, gene sequences comparison and analysis are the most basic routines. With the considerable increase of sequences to analyze, we need more and more computing power as well as efficient solution to manage data.

In this work, we have investigated the advantages of using a new Desktop Grid middleware BITDEW, designed for large scale data management. Our contribution is two-fold: firstly, we introduce a data-driven Master/Slave programming model and we present an implementation of BLAST over BITDEW following this model, secondly, we present extensive experimental and simulation results which demonstrate the effectiveness and scalability of our approach. We evaluate the benefit of multi-protocol data distribution to achieve remarkable speedups, we report on the ability to cope with highly volatile environments with relative performance degradation, we show the benefit of data replication in Grid with heterogeneous resource performance and we evaluate the combination of data fault tolerance and data replication when computing on volatile resources.

6.2.10. MapReduce programing model for Desktop Grid

Since its introduction in 2004 by Google, MapReduce has become the programming model of choice for processing large data sets. MapReduce borrows from functional programming, where a programmer can define both a Map task that maps a data set into another data set, and a Reduce task that combines intermediate outputs into a final result. Although MapReduce was originally developed for use by web enterprises in large data-centers, this technique has gained a lot of attention from the scientific community for its applicability in large parallel data analysis (including geography, high energy physics, genomics, etc..).

During 2009, we have started an implementation of the MapReduce programming model on Desktop Grid using the BITDEW middleware. Although this research addresses many issues, such as efficient scheduling of data and tasks, distributed result certification, large scale collective communication (broadcast and reduction) on volatile resources, early experiments with the prototype are being done on Grid5K to evaluate the performance of our implementation. We expect the first results during 2010.

6.2.11. Bridging Grid and Desktop Grid

Service grids and desktop grids are both promoted by their supportive communities as effective solutions for providing huge computing power. Little work, however, has been undertaken to blend these two technologies together in an effort to create one vast and seamless pool of resources. In the context of the EDGeS FP7 infrastructures project, entitled Enabling Desktop Grids for e-Science (EDGeS), we collaborate to build technological bridges to facilitate service and desktop grids interoperability. Within the consortium, we are leader of the JRA1 work package which provides the software to bridge Desktop Grids and Service Grids. In past work, we have given a detailed presentation of the BOINC to EGEE bridge, and we addressed the security issues when bridging Service Grids with Desktop Grids.

In 2009, we have extended the EDGeS bridge so that EGEE users can get access to additional resources provided by XTREMWEB-HEP Desktop Grids. We have built a new public XTREMWEB Desktop Grid called EDGeS@Home, which allows the general public to donate their idle time to EGEE users by executing EGEE applications in a way similar to what BOINC does. Finally, we have set up two bridges which connect the EDGeS VO to three different XTREMWEB-HEP based Desktop Grids running at the University Paris-XI. We plan to extend this test and production infrastructure to Grid5K.

6.2.12. Sandboxing for Desktop Grid

In this work, we have investigated methods and mechanisms that enable the use of virtual machines as part of a security infrastructure for Desktop Grid clients to provide a sandbox for running (untrusted) applications.

Desktop Grids harvest the computing power of idle desktop computers whether these are volunteer or deployed at an institution. Allowing foreign applications to run on these resources requires the sender of the application to be trusted, but trust in goodwill is never enough. An efficient solution is to provide a secure isolated execution environment ("sandbox"), which does not constrain any additional burden neither on administrators nor on users. Currently Desktop Grids do not provide such facility. We defined and analyzed the requirements for any platform independent and transparent sandbox for Desktop Grids. We designed a prototype, which we built based on our findings and we give a performance evaluation.

6.2.13. Meta-Scheduling and Task Reallocation in a Grid Environment

Parallel resources in a grid are generally accessed through a batch system which both schedules and reserves the resources in accordance to its scheduling policy. Each batch system has its own scheduling algorithm which constructs the schedule with the available task information (number of requested processors, walltime, for example) at submission time. However, walltime is generally over-estimated. This does not necessarily have consequences at the local level in terms of resource utilization if techniques of backfilling are used, but at the grid level, the meta-schedule built by the grid middleware may not be the best anymore according to the metric to optimize.

Thus, we have explored the possibility to migrate grid-submitted jobs which are still in the waiting queue of batch systems with which the grid middleware is discussing. Several heuristics have been tested, among which well-known heuristics and a lot of experiments have been simulated using real life batch traces. Work is still in progress with automatically-tuned parallel applications.

We aim to decide if migrating waiting tasks is interesting in terms of optimizing some metrics (quantify) and to test which mechanisms have to be involved in a real implementation in a grid middleware, since the implementation and maintenance cost of that kind of code may be a considerable drawback if the gain is too small.

6.2.14. Enabling Distributed Computation and Fault-Tolerance Among Stigmergic Robots

We investigate avenues for the exchange of information (explicit communication) among deaf and dumb mobile robots scattered in the plane. We introduce the use of movement-signals (analogously to flight signals and bee waggle) as a mean to transfer messages, enabling the use of distributed algorithms among robots. We propose one-to-one deterministic movement protocols that implement explicit communication among asynchronous robots. We first show how the movements of robots can provide implicit acknowledgment in asynchronous systems. We use this result to design one-to-one communication among a pair of robots. Then, we propose two one-to-one communication protocols for any system of $n \geq 2$ robots. The former works for robots equipped with observable IDs that agree on a common direction (sense of direction). The latter enables one-to-one communication assuming robots are devoid of any observable ID or sense of direction. All three protocols (for either two or any number of robots) assume that no robot remains inactive forever. However, they cannot avoid that the robots move either away from, or closer to, each other, by the way requiring robots with an infinite visibility. We also show how to overcome these two disadvantages.

These protocols enable the use of distributed algorithms based on message exchanges among swarms of Stigmergic robots. They also allow robots to be equipped with means of communication to tolerate faults in their communication devices.

6.3. Parallel Sparse Direct Solvers and Combinatioral Scientific Computing

Participants: Maurice Brémond, Indranil Chowdhury, Guillaume Joslin, Jean-Yves L'Excellent, Bora Uçar.

6.3.1. Extension, support and maintenance of the software package MUMPS

This year, we have pursued work to add functionalities and improve the MUMPS software package. For example, the parallel analysis which we worked on last year has been made available by default in the public releases of the package, and some of the research work on out-of-core issues has been integrated and validated. As usual, we have had strong interactions with many users (e.g., in the context of the Samtech or Solstice

projects, but also through informal collaborations), and this has led us to work on the following points: (i) 64-bit integers to address larger memories; (ii) improvement of load balance and better scalability on specific classes of matrices from EDF and from the French-Israeli Multicomputing project; (iii) more flexible interface from the memory usage point of view, compatibility of compressed orderings with more ordering packages, various performance improvements and bug corrections.

To conclude this section, notice that an action of technological development funded by INRIA (ADTMUMPS) has just started which should significantly help improving software engineering aspects, documentation, and developers' tools to validate and experiment the package.

6.3.2. Multithreading

The aim of this starting work is to multithread several parts of the MUMPS solver in order to utilize modern multi-core machines more effectively by adding an OpenMP layer on top of the already existing MPI implementation. In addition, using threaded BLAS libraries (such as Goto, MKL and ACML) can provide significant speedups for the BLAS operations within MUMPS. Pure shared memory codes do not exhibit linear speedups with increasing number of cores, and they tend to saturate. The idea here is to increase parallelism by mixing OpenMP along with the MPI processes. Till now, we have identified the bottlenecks of the serial code using profilers like TAU and VTUNE, and have experimented performance improvements by putting OpenMP directives for maximal utilization of the available cores. Significant speedup was observed during the assembly and memory stacking phases, some other key areas like pivot search and solution operations are still being investigated. It has been found that the mixed MPI and OpenMP strategy works well for large unsymmetric cases, where we achieve almost 6 times speedup using 8 cores. However for symmetric cases a pure MPI run on the available cores seems to show optimum performance, the reason for which is not clear. In the next few months we intend to wrap up the OpenMP work and set guidelines for the users who are interested in shared memory implementation of MUMPS.

6.3.3. Exact algorithms for a task assignment problem

We consider the following task assignment problem. Communicating tasks are to be assigned to heterogeneous processors interconnected with a heterogeneous network. The objective is to minimize the total sum of the execution and communication costs. The problem is NP-hard. We present an exact algorithm based on the well-known A* search. We report simulation results over a wide range of parameters where the largest solved instance contains about three hundred tasks to be assigned to eight processors.

6.3.4. On the block triangular form of symmetric matrices

We present some observations on the block triangular form (btf) of structurally symmetric, square, sparse matrices. If the matrix is structurally rank deficient, its canonical btf has at least one underdetermined and one overdetermined block. We prove that these blocks are transposes of each other. We further prove that the square block of the canonical btf, if present, has a special fine structure. These findings help us recover symmetry around the anti-diagonal in the block triangular matrix. The uncovered symmetry helps us to permute the matrix in a special form which is symmetric along the main diagonal while exhibiting the blocks of the original btf. As the square block of the canonical btf has full structural rank, the observation relating to the square block applies to structurally nonsingular, square symmetric matrices as well.

6.3.5. On two-dimensional sparse matrix partitioning: Models, methods, and a recipe

We consider two-dimensional partitioning of general sparse matrices for parallel sparse matrix-vector multiply operation. We present three hypergraph-partitioning based methods, each having unique advantages. The first one treats the nonzeros of the matrix individually and hence produces fine-grain partitions. The other two produce coarser partitions, where one of them imposes a limit on the number of messages sent and received by a single processor, and the other trades that limit for a lower communication volume. We also present a thorough experimental evaluation of the proposed two-dimensional partitioning methods together with the hypergraph-based one-dimensional partitioning methods, using an extensive set of public domain matrices. Furthermore, for the users of these partitioning methods, we present a partitioning recipe that chooses one of the partitioning methods according to some matrix characteristics.

6.3.6. On the scalability of hypergraph models for sparse matrix partitioning

We investigate the scalability of the hypergraph-based sparse matrix partitioning methods with respect to the increasing sizes of matrices and number of nonzeros. We propose a method to rowwise partition the matrices that correspond to the discretization of two-dimensional domains with the five-point stencil. The proposed method obtains perfect load balance and achieves very good total communication volume. We investigate the behaviour of the hypergraph-based rowwise partitioning method this with respect to the proposed method, in an attempt to understand how scalable the former method is. In another set of experiments, we work on general sparse matrices under different scenarios to understand the scalability of some other hypergraph-based partitioning methods.

7. Contracts and Grants with Industry

7.1. Contract with SAMTECH, 2008-2010

INRIA and INPT-IRIT have signed a new contract with the company Samtech S.A. (Belgium). Samtech develops the finite element software package SAMCEF, which uses our parallel sparse direct solver MUMPS as one of the internal solvers. The goal of this contract is to improve the memory usage of MUMPS, and to offer the possibility to address a larger amount of memory. We will also study how to use memory already allocated by SAMCEF instead of having the solver allocate its own memory. Finally we also plan to study how performance can be improved on Samtech problems by allowing the forward substitution step to be performed simultaneously with the matrix factorization. This last point is particularly interesting in the case of out-of-core executions.

The contract is 24-month long, and the new functionalities developed in MUMPS for this contract will be made available in a future public release of the package.

In Lyon, J.-Y. L'Excellent is the principal investigator and B. Uçar participates to this contract.

8. Other Grants and Activities

8.1. Regional Projects

8.1.1. Pôle Scientifique de Modélisation Numérique (PSMN)

This federation of laboratories aims at sharing the parallel machines from ENS Lyon/PSMN and experiences of parallelization of applications.

J.-Y. L'Excellent participates to this project.

8.1.2. MUSINE: Franche-Comté: conception, validation et pilotage de la micro-usine multi-cellulaire (2007-2009)

The aim of this project is to design the information model and management (scheduling) part of a micro-factory composed of cells. Each cell contains a set of micro-robots which manipulate micro-products (about 10^{-5} meters). The project is in collaboration with the LAB (Laboratoire d'Automatique de Besançon).

L. Philippe leads the MUSINE project and J.-M. Nicod participates to it.

8.1.3. Projet "Calcul Hautes Performances et Informatique Distribuée"

E. Caron leads (with C.. Prudhomme from LJK, Grenoble) the "Calcul Hautes Performances et Informatique Distribuée" project of the cluster "Informatique, Signal, Logiciels Embarqués". Together with several research laboratories from the Rhône-Alpes region, we initiate collaborations between application researchers and distributed computing experts. A Ph.D. thesis (J.-S. Gay) focuses on the scheduling problems for physics and bioinformatic applications.

Y. Caniou, E. Caron, F. Desprez, J.-Y. L'Excellent, J.-S. Gay, and F. Vivien participate to this project.

8.2. National Contracts and Projects

8.2.1. ANR grant: Stochagrid (Scheduling algorithms and stochastic performance models for workflow applications on dynamic Grid platforms), 3 years, ANR-06-BLAN60192-01, 2007-2010

In the second year of the project (2009), we have investigated timed-Petri nets to model the mapping of workflows with stage replication, and we have succeeded in deriving an optimal polynomial algorithm to compute the period in the bounded multi-port model with overlap. Quite interestingly, the period is no longer the bottleneck resource, the critical path becomes more complex. We have extended these results to probabilistic workflows in which execution and communication times follow random variable laws, such as exponential laws. Certain parts of the work was conducted in collaboration with Bruno Gaujal (LIG Grenoble). We have also investigated several multi-criteria algorithms and heuristics, and we hired a post-doctoral student to conduct work on the implementation of robust scheduling algorithms.

The project is entirely conducted within the GRAAL team by A. Benoit and Y. Robert.

8.2.2. ANR grant CICG-05-11: LEGO (League for Efficient Grid Operation), 3 years, 2006-2009

The aim of this project is to provide algorithmic and software solutions for large scale architectures; our focus is on performance issues. The software component provides a flexible programming model where resource management issues and performance optimizations are handled by the implementation. On the other hand, current component technology does not provide adequate data management facilities, needed for large data in widely distributed platforms, and does not deal efficiently with dynamic behaviors. We choose three applications: ocean-atmosphere numerical simulation, cosmological simulation, and sparse matrix solver. We propose to study the following topics: Parallel software component programming; Data sharing model; Network-based data migration solution; Co-scheduling of CPU, data movement and I/O bandwidth; High-perf. network support. The Grid'5000 platform provides the ideal environment for testing and validation of our approaches.

E. Caron is leading the project, which comprises six teams: GRAAL/LIP (Lyon), PARIS/IRISA (Rennes), RUNTIME/LaBRI (Bordeaux), ENSEEIHT/IRIT (Toulouse), CERFACS (Toulouse) and CRAL/ENS-Lyon (Lyon). A. Amar, R. Bolze, Y. Caniou, F. Desprez, J.-S. Gay and C. Tedeschi also participate to this project.

The project has ended in June 2009.

8.2.3. ANR grant ANR-06-CIS-010: SOLSTICE (SOlveurs et simulaTIon en Calcul Extrême), 3 years, 2007-2009

The objective of this project is to design and develop high-performance parallel linear solvers that will be efficient to solve complex multi-physics and multi-scale problems of very large size (10 to 100 millions of equations). To demonstrate the impact of our research, the work produced in the project will be integrated in real simulation codes to perform simulations that could not be considered with today's technologies. This project also comprises *LaBRI* (coordinator), CERFACS, INPT-IRIT, CEA-CESTA, EADS-CCR, EDF R&D, and CNRM. We are more particularly involved in tasks related to out-of-core factorization and solution, parallelization of the analysis phase of sparse direct solvers, rank detection, hybrid direct-iterative methods and expertise site for sparse linear algebra.

Indranil Chowdhury, Guillaume Joslin, Jean-Yves L'Excellent and Bora Uçar participate to this project.

8.2.4. ANR grant ANR-06-MDCA-009: Gwendia (Grid Workflow Efficient Enactment for Data Intensive Applications), 3 years, 2007-2009

The objective of the Gwendia[5] project is to design and develop workflow management systems for applications involving large amounts of data. It is a multidisciplinary project involving researchers in computer science (including GRAAL) and in life science (medical imaging and drug discovery). Our work consists in designing algorithms for the management of several workflows in distributed and heterogeneous platforms and to validate them within DIET on the Grid'5000 platform.

8.2.5. ANR grant: COOP (Multi Level Cooperative Resource Management), 3 years, ANR-09-COSI-001-01, 2009-2012

The main goals of this project are to set up such a cooperation as general as possible with respect to programming models and resource management systems and to develop algorithms for efficient resource selection. In particular, the project targets the SALOME platform and GRID-TLSE expert-site (http://gridtlse.org/) as example of programming models, and Marcel/PadicoTM, DIET and XtreemOS as examples of multithread scheduler/communication manager, grid middleware and distributed operating systems.

The project is led by Christian Perez.

8.2.6. ANR JCJC: Clouds@Home (Cloud Computing over Unreliable, Shared Resources), 4 years, ANR-09-JCJC-0056-01, 2009-2012

Recently, a new vision of cloud computing has emerged where the complexity of an IT infrastructure is completely hidden from its users. At the same time, cloud computing platforms provide massive scalability, 99.999% reliability, and speedy performance at relatively low costs for complex applications and services. This project, lead by D. Kondo from INRIA MESCAL investigates the use of cloud computing for large-scale and demanding applications and services over unreliable resources. In particular, we target volunteered resources distributed over the Internet. In this project, G. Fedak leads the Data management task (WP3).

8.2.7. ADTMUMPS, 3 years, 2009-2012

ADTMUMPS is an action of technological development funded by INRIA. This project gives support for 24 men x months of young engineer ("ingénieur jeune diplômé"). A permanent engineer from INRIA/SED also works on the project (Maurice Brémond, 30 % on the project). One goal of the project is to improve daily work of MUMPS developers by improving the software engineering aspects, by developing non-regression tests and drivers to experiment the package. This project is in collaboration with ENSEEIHT-IRIT.

8.2.8. ADT ALADDIN

ALADDIN is an INRIA action of technological development for "A LArge-scale DIstributed and Deployable INfrastructure" which aim is to manage the Grid'5000 experimental platform. Frédéric Desprez is leading this project (with David Margery from Rennes as the Technical Director).

8.3. European Contracts and Projects

8.3.1. Marie Curie Action – IOF – MetagenoGrids

In the scope of the associated-team described in the next section, Frédéric Vivien was on sabbatical at the University of Hawai'i at Manoā for one year, from July 17, 2008 until July 16, 2009. This sabbatical was in part funded by a Marie Curie Action – International Outgoing Fellowship from the European Commission.

8.3.2. ERCIM WG CoreGRID (2009-2011)

Following the success of the NoE CoreGRID, an ERCIM WG was started in 2009, leaded by F. Desprez. This working group gathers 31 research teams from all over europe working on Grids, Service oriented architectures and Clouds.

[5] http://gwendia.polytech.unice.fr/doku.php

A workshop on Grids, P2P and Service computing was organized in conjunction with EuroPAR 2009, Delft, August, 2009.

8.3.3. EU FP7 project EDGeS: Enabling Desktop Grids for e-Science (2008-2009)

This project is lead by P. Kacsuk, and involves the following partners : SZTAKI, INRIA, CIEMAT, Fundecyt, University of Westminster, Cardiff University, University of Coimbra. Grid systems are currently being used and adopted by a growing number of user groups and diverse application domains. However, there still exist many scientific communities whose applications require much more computing resources than existing Grids like EGEE can provide. The main objective of this project is to interconnect the existing EGEE Grid infrastructure with existing Desktop Grid (DG) systems like BOINC or XTREMWEB in a strong partnership with EGEE. The interconnection of these two types of Grid systems will enable more advanced applications and provide extended compute capabilities to more researchers. In this collaboration G. Fedak represents the GRAAL team and is responsible for JRA1: Service Grids-Desktop Grids Bridges Technologies and is involved in JRA3 : Data Management, as well as NA3 : Standardization within the OGF group.

8.4. International Contracts and Projects

8.4.1. France-Berkeley Fund Award (2008-2009)

In the framework of the France-Berkeley Fund, we have been awarded a research grant to enable an exchange program involving both young and confirmed scientists. The project focused on massively parallel solvers for large sparse matrices and reinforced the collaboration initiated by E. Agullo when he was a member of the GRAAL team. At LIP, J.-Y. L'Excellent and B. Uçar participated to this project in 2009. On the French side, this project also involves ENSEEIHT-IRIT, *LaBRI*, and CERFACS.

8.4.2. French-Israeli project "Multicomputing" (2009-2010)

This project aims at improving the scalability of state-of-the-art computational fluid dynamics calculations by the use of state-of-the-art numerical linear algebra approaches. It mainly involves Tel Aviv University and ENSEEIHT-IRIT (Toulouse), where Alfredo Buttari is coordinator for the French side. In GRAAL, I. Chowdhury, J.-Y. L'Excellent, and B. Uçar participate to this project.

8.4.3. REDIMPS (2007-2009)

REDIMPS (Research and Development of International Matrix Prediction System) is a project funded by the Strategic Japanese-French Cooperative Program on "Information and Communications Technology including Computer Science" with the CNRS and the JST. The goal of this international collaboration is building an international sparse linear equation solver expert site. Among the objectives of the project, one resides in the cooperation of the TLSE partners and the JAEA in the testing, the validation and the promotion of the TLSE system that is currently released. JAEA, who is one of the leading institute and organization of Japanese HPC, is studying high-performance numerical simulation methods on novel supercomputers, and is expecting to find the best linear solver within this collaboration. By integrating knowledge and technology of JAEA and TLSE partners, it is expected that we will achieve the construction of an international expert system for sparse linear algebra on an international grid computing environment.

Thanks to additional funding from INRIA's "explorateur" program, Y. Caniou spent one month two times at the Japan Atomic Energy Agency in Tokyo, Japan. He worked on the AEGIS-DIET Grid system interoperability.

Yves Caniou, Eddy Caron, Frédéric Desprez, and Jean-Yves L'Excellent participate to this project.

8.4.4. CNRS-USA grant SchedLife, University of Hawai'i (2007-2009)

We have been awarded a CNRS grant in the framework of the CNRS/USA funding scheme, which runs for three years starting in 2007. The collaboration is done with the Concurrency Research Group (CoRG) of Henri Casanova, and the Bioinformatics Laboratory (BiL) of Guylaine Poisson of the Information and Computer Sciences Department, of the University of Hawai'i at Manoā, USA.

The SchedLife project targets the efficient scheduling of large-scale scientific applications on clusters and Grids. To provide context for this research, we focus on applications from the domain of bioinformatics, in particular comparative genomics and metagenomics applications, which are of interest to a large user community today. So far, applications (in bioinformatics or other fields) that have been successfully deployed at a large scale fall under the "independent task model": they consist of a large number of tasks that do not share data and that can be executed in any order. Furthermore, many of these application deployments rely on the fact that the application data for each task is "small", meaning that the cost of sending data over the network can be ignored in the face of long computation time. However, both previous assumptions are not valid for all applications, and in fact many crucial applications, such as the aforementioned bioinformatics applications, require computationally dependent tasks sharing very large data sets.

In our previous collaborations, we have tackled the issue of non-negligible network communication overheads and have made significant contributions. For instance, we have designed strategies that rely on the notions of steady-state scheduling (i.e., attempting to maximize the number of tasks that complete per time unit, in the long run) and/or divisible load scheduling (i.e., approximate the discrete workload that consists of individual tasks as a continuous workload). These strategies provide powerful means for rethinking the deployment and the scheduling of independent task applications when network communication can be a bottleneck. However, the target applications in this project cannot benefit from these strategies directly and will require fundamental advances. This project aims to build upon and go beyond our past collaborations, with two main research thrusts:

- Scheduling of applications with data requirements. We consider applications that require possibly multiple data files that need to be shared by multiple application tasks. These files may be extremely large (e.g., millions of genomic sequences) and may need to be updated frequently (e.g., when new sequences are identified). We must then ensure that file access is not a bottleneck.
- Scheduling of multiple concurrent applications. We also plan to study the scheduling for multiple applications, i.e., launched by different (most likely competing) users. We then aim to orchestrate computation and communication in order to have the best aggregate performance. This is a difficult problem, first in order to define a good performance metric, and then to maximize this performance metric in a tractable way.

A. Benoit, E. Caron, F. Desprez, Y. Robert and F. Vivien participate to this project.

8.4.5. Associated-team MetagenoGrid (2008-2010)

This associated-team involves the exact same persons, and covers the same subject, as the CNRS-USA grant SchedLife described above.

8.4.6. CNRS délégation of Yves Caniou (2009-2010)

Yves Caniou has obtained a CNRS delegation for the scholar year 2009-2010. He is now working at the CNRS Japan-French Laboratory in Informatics (JFLI) supervised by Philippe Codognet. The JFLI is located in Tokyo, Japan, and is composed of the Tokyo University, Université Pierre et Marie-Curie (UPMC), the Keio University, the CNRS, the NII partnership.

9. Dissemination

9.1. Scientific Missions

Open Grid Forum. The objective of the Open Grid Forum working group on "Grid Remote Procedure Call" (GridRPC) is to define a standard for this way to use Grid resources. E. Caron is co-chair of this OGF working group. E. Caron, Y. Caniou and F. Desprez participated to the elaboration of a GridRPC Data Management API.

9.2. Edition and Program Committees

Anne Benoit is the Program Chair of the 19th International Heterogeneity in Computing Workshop, HCW 2010, to be held in Atlanta, USA, April 2010, in conjunction with IPDPS 2010. She co-organized the 6th International Workshop on aPplications of declArative and object-oriented Parallel Programming (PAPP 2009), Baton Rouge, Louisiana, USA, May 2009; she is co-organizing the 7th edition of the workshop PAPP 2010 in Amsterdam, The Netherlands, May 2010.

A. Benoit was a member of the program committee of ICCS 2009, HPCC 2009, ISPDC 2009, ISCIS 2009, TCPP PhD Forum 2009. She is a member of the program committee of ICCS 2010, IPDPS 2010 and APDCM 2010.

Yves Caniou is a member of the program committee of Heterogeneous Computing Workshop 2009 and 2010, and of the ICCSA 2009 and 2010 conferences.

Eddy Caron was a member of the program committee of RENPAR 2009, PDP 2009, HCW 2009, MGC 2009, ISPA 2009, HotP2P 2009, GridPeer 2009 and SSS 2009.

He was co-chair of Tutorial Sesion for CCGRID'2008, and is co-chair of Grid-RPC group in the OGF (Open Grid Forum).

In november 2009, E. Caron organized the SSS'09 conference in Lyon that gathered 130 delegates

He was co-organizer of INRIA Booth at SuperComputing 2009.

He is a co-funder of the SysFera startup company.

Frédéric Desprez is member of the EuroPar Advisory board and the editorial board of "Scalable Computing: Practice and Experience" (SCPE).

F. Desprez participated to the program committees of CLADE'09, VecPar'09, HPDC'09 (IEEE Int. Symp. on High-Performance Distributed Computing), CCGRID'09, PCGrid'09, chair of the algorithms track for the poster session of Supercomputing'09, Modern computer tools for the biosciences'09, Third Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid'09), DEPEND'09, SSS'09, 8th International Conference on Service Oriented Computing (ICSOC '09) joint with ServiceWave'09.

In september 2009, F. Desprez organized the ParCo conference in Lyon that gathered 160 delegates [6]. In August 2009, he organized the workshop on Grids, P2P and Service Computing joint with EuroPAR'09 [7].

He is a co-funder of the SysFera startup company.

Gilles Fedak was the program chair of the PCGRID workshop associated with IPDPS'09. He co-chairs 2 workshops PCGRID'10 and MAPREDUCE'10 associated respectively with CCGRID (Melbourne Australia, 2010) and HPDC (Chicago, USA, 2010). He is the track co-chair for the High-speed Distributed Systems and Grids (HDSG) track in the 19th IEEE International Conference on Computer Communications and Networks (ICCCN), Zurich, Switzerland, August 2010.

He was a member of the program committees of the following conferences and workshops : CoreGrid'09, (Delft, Nederlands, 2009), CDUR'09 (Montreal, Canada, 2009), CCGRID'09 (Shanghai, China, 2009), ServP2P'09 (Shanghai, China, 2009), RenPar'09 (Toulouse, 2009).

Jean-Yves L'Excellent is a member of the program committee of Vecpar'10 (Berkeley, California).

Loris Marchal was a member of the program committee of ISPDC 2009. He is co-editor (with Frédéric Vivien) of a special issue of the Parallel Computing journal following a workshop organized in 2008.

Christian Pérez was a member of the program committee of AINA (Bradford, UK, May 26-29, 2009), CCGRID (Shangai, China, May 18-21, 2009), CBHPC (Portland, Oregon, USA, 15-16 November 2009), HPCC (Seoul, Korea, June 25-27, 2009), ParCo (Lyon, France, September 1-4, 2009) and RenPar (Toulouse, France, 9-11 September 2009). He was publicity co-chair of HPCS (Leipzig, Germany June 21-24, 2009).

[6] http://www.parco.org/
[7] http://www.coregrid.net/mambo/content/view/784/435/

He is a member of the program committee of FMMC (Heidelberg Academy of Sciences, Germany, March 17-19, 2010), Vecpar'10 (Berkeley, CA, USA, 22-25 June 2010). He is a member of the Steering Committee of CBHPC.

C. Pérez serves as expert for evaluating proposal to the ANR Cosinus 2009 call and for evaluating a Thalès PhD CIFRE candidate.

Yves Robert is a member of the editorial board of the *International Journal of High Performance Computing Applications* (Sage Press), and of the *Journal of Computational Science* (Elsevier).

Y. Robert was Program Chair of the IEEE TCPP PhD Forum in Rome (in conjunction with IPDPS'09). He also was Program Chair of ISPDC'09, the 8th International Symposium on Parallel and Distributed Computing that took place in Lisbon in July 2009. He will be the Program Vice-Chair of HiPC'2010, track Algorithms and Applications.

Yves Robert is a member of the Steering Committee of HCW (IEEE Workshop on Heterogeneity in Computing) of IPDPS (IEEE Int. Parallel and Distributed Symposium), and of HeteroPar (Int. Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms).

Following the 35th (French) Spring school in theoretical computer science (EPIT) that they organized in June 2007, Y. Robert and F. Vivien have edited a book on *Introduction to scheduling*. This book has been published in 2009 by Chapman and Hall/CRC Press.

Bora Uçar was a member of the program committee of Fifteenth International Conference on Parallel and Distributed Systems (ICPADS'09), Shenzhen, China, December 9–11, 2009; of the 24th International Symposium on Computer and Information Sciences (ISCIS 2009), METU Northern Cyprus Campus, September 14–16, 2009; of IPDPS 2009 TCPP PhD Forum.

B. Uçar organized a mini-symposium entitled "Parallel sparse matrix computations and enabling algorithms"as a part of SIAM Conference on Computational Science & Engineering (CSE09), March 2–6, 2009, Miami, Florida, USA.

Frédéric Vivien is an associate editor of *Parallel Computing*.

F. Vivien was a member of the program committee of EuroPDP 2010, Pisa, Italy, February 2010, of NPC 2009, Gold coast, Australia, October 19-21, 2009, of CCGrid 09, Shangai, China, May 18-21, 2009 (only as member of the backup team), of the *Workshop on Scheduling for Parallel Computing*, Wroclaw, Poland, September 2009; of RenPar 2009, Toulouse, France, September 2009, of ISPDC'2009, Lisbon, Portugal, June 30 - July 4, 2009, and of EuroPDP 2009, Weimar, Germany, February 2009.

9.3. Administrative and Teaching Responsibilities

9.3.1. Teaching Responsibilities

Licence d'Informatique Fondamentale at ENS Lyon. Anne Benoit is responsible of the 3rd year students on fundamental computer science at ENS Lyon.

Jean-Yves L'Excellent and Bora Uçar offered CR09-Sparse matrix computations lecture series in the Master d'Informatique Fondamentale at ENS Lyon.

Eddy Caron and Christian Perez gave a series of lectures entitled *CR11-Grid Computing* in the Master d'Informatique Fondamentale at ENS Lyon.

Master in Computer Science at Université de Franche Comté. Jean-Marc Nicod is the head of the Master in Computer Science of Université de Franche-Comté. L. Philippe and J.-M. Nicod participate to this master and give advanced classes related to distributed computing and distributed algorithms.

10. Bibliography

Major publications by the team in recent years

[1] P. R. AMESTOY, I. S. DUFF, J. KOSTER, J.-Y. L'EXCELLENT. *A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling*, in "SIAM Journal on Matrix Analysis and Applications", vol. 23, n° 1, 2001, p. 15-41.

[2] C. BANINO, O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, Y. ROBERT. *Scheduling strategies for master-slave tasking on heterogeneous processor platforms*, in "IEEE Trans. Parallel Distributed Systems", vol. 15, n° 4, 2004, p. 319-330.

[3] O. BEAUMONT, L. CARTER, J. FERRANTE, A. LEGRAND, L. MARCHAL, Y. ROBERT. *Centralized versus distributed schedulers for multiple bag-of-task applications*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n° 5, 2008, p. 698-709.

[4] O. BEAUMONT, H. CASANOVA, A. LEGRAND, Y. ROBERT, Y. YANG. *Scheduling divisible loads on star and tree networks: results and open problems*, in "IEEE Trans. Parallel Distributed Systems", vol. 16, n° 3, 2005, p. 207-218.

[5] A. BENOIT, V. REHN-SONIGO, Y. ROBERT. *Replica placement and access policies in tree networks*, in "IEEE Trans. Parallel Distributed Systems", vol. 19, n° 12, 2008, p. 1614-1627.

[6] E. CARON, F. DESPREZ. *DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid*, in "International Journal of High Performance Computing Applications", vol. 20, n° 3, 2006, p. 335-352.

[7] F. DESPREZ, J. DONGARRA, A. PETITET, C. RANDRIAMARO, Y. ROBERT. *Scheduling block-cyclic array redistribution*, in "IEEE Trans. Parallel Distributed Systems", vol. 9, n° 2, 1998, p. 192-205.

[8] F. DESPREZ, F. SUTER. *Impact of Mixed-Parallelism on Parallel Implementations of Strassen and Winograd Matrix Multiplication Algorithms*, in "Concurrency and Computation: Practice and Experience", vol. 16, n° 8, July 2004, p. 771–797.

[9] A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Constructing Memory-minimizing Schedules for Multifrontal Methods*, in "ACM Transactions on Mathematical Software", vol. 32, n° 1, 2006, p. 17–32.

[10] A. LEGRAND, A. SU, F. VIVIEN. *Minimizing the stretch when scheduling flows of divisible requests*, in "Journal of Scheduling", vol. 11, n° 5, 2008, p. 381-404.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] A. BENOIT. *Scheduling pipelined applications: models, algorithms and complexity*, École normale supérieure de Lyon, July 2009, Habilitation à diriger des recherches.

[12] M. GALLET. *Steady-State Scheduling of Workflow Applications onto Heterogeneous Platforms*, École Normale Supérieure de Lyon, October 2009, Ph. D. Thesis.

[13] V. REHN-SONIGO. *Multi-criteria Mapping and Scheduling of Workflow Applications onto Heterogeneous Platforms*, École Normale Supérieure de Lyon, July 2009, Ph. D. Thesis.

Articles in International Peer-Reviewed Journal

[14] E. AGULLO, A. GUERMOUCHE, J.-Y. L'EXCELLENT. *Reducing the I/O Volume in Sparse Out-of-core Multifrontal Methods*, in "SIAM Journal on Scientific Computing", 2010, to appear.

[15] F. BAUDE, C. DENIS, C. DALMASSO, M. DANELUTTO, V. GETOV, L. HENRIO, C. PÉREZ. *GCM: A Grid Extension to Fractal for Autonomous Distributed Components*, in "Special Issue of Annals of Telecommunications: Software Components – The Fractal Initiative", vol. 64, n^{o} 1, 2009, p. 5–24 UK IT .

[16] A. BENOIT, M. HAKEM, Y. ROBERT. *Contention awareness and fault tolerant scheduling for precedence constrained tasks in heterogeneous systems*, in "Parallel Computing", 2009, To appear.

[17] A. BENOIT, M. HAKEM, Y. ROBERT. *Multi-criteria scheduling of precedence task graphs on heterogeneous platforms*, in "The Computer Journal", 2009, To appear.

[18] A. BENOIT, H. KOSCH, V. REHN-SONIGO, Y. ROBERT. *Multi-criteria scheduling of pipeline workflows (and application to the JPEG encoder)*, in "Int. Journal of High Performance Computing Applications", vol. 23, n^{o} 2, 2009, p. 171-187 DE .

[19] A. BENOIT, L. MARCHAL, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Scheduling concurrent bag-of-tasks applications on heterogeneous platforms*, in "IEEE Transactions on Computers", 2009, To appear.

[20] A. BENOIT, Y. ROBERT. *Complexity results for throughput and latency optimization of replicated and data-parallel workflows*, in "Algorithmica", 2009, To appear.

[21] A. BENOIT, E. THIERRY, Y. ROBERT. *On the complexity of mapping linear chain applications onto heterogeneous platforms*, in "Parallel Processing Letters", vol. 19, n^{o} 3, 2009, p. 383-397.

[22] V. BERTIS, R. BOLZE, F. DESPREZ, K. REED. *From Dedicated Grid to Volunteer Grid: Large Scale Execution of a Bioinformatics Application*, in "Journal of Grid Computing", 2009, To appear.

[23] E. CARON, F. DESPREZ, C. TEDESCHI, F. PETIT. *Snap-Stabilizing Prefix Tree for Peer-to-Peer Systems*, in "Parallel Processing Letters", 2009, To appear.

[24] S. DAHAN, L. PHILIPPE, J.-M. NICOD. *The Distributed Spanning Tree Structure*, in "IEEE Trans. Parallel Distributed Systems", 2009, To appear.

[25] I. S. DUFF, B. UÇAR. *On the block triangular form of symmetric matrices*, in "SIAM Review", 2010, To appear GB .

[26] G. FEDAK, H. HE, F. CAPPELLO. *A Data Management and Distribution Service with Multi-Protocol and Reliable File Transfer*, in "Journal of Network and Computer Applications", vol. 32, n^{o} 5, September 2009, p. 961–975.

[27] K. KAYA, B. UÇAR. *Exact algorithms for a task assignment problem*, in "Parallel Processing Letters", vol. 19, n^{o} 3, 2009, p. 451-465 TR .

[28] F. SOURBIER, S. OPERTO, J. VIRIEUX, P. R. AMESTOY, J.-Y. L'EXCELLENT. *FWT2D: a massively parallel program for frequency-domain Full-Waveform Tomography of wide-aperture seismic data – Part 1: Algorithm*, in "Computer and Geosciences", vol. 35, n^{o} 3, 2009, p. 487-495.

[29] F. SOURBIER, S. OPERTO, J. VIRIEUX, P. R. AMESTOY, J.-Y. L'EXCELLENT. *FWT2D: a massively parallel program for frequency-domain Full-Waveform Tomography of wide-aperture seismic data – Part 2: numerical examples and scalability analysis*, in "Computer and Geosciences", vol. 35, n^{o} 3, 2009, p. 496-514.

[30] E. URBAH, P. KACSUK, Z. FARKAS, G. FEDAK, G. KECSKEMETI, O. LODYGENSKY, A. C. MAROSI, Z. BALATON, G. CAILLAT, G. GOMBAS, A. KORNAFELD, J. KOVACS, H. HE, R. LOVAS. *EDGeS: Bridging EGEE to BOINC and XtremWeb*, in "Journal of Grid Computing", vol. 7, n^{o} 3, September 2009, p. 335–354 HU GB PT ES NL .

[31] ÜMIT. V. ÇATALYÜREK, C. AYKANAT, B. UÇAR. *On two-dimensional sparse matrix partitioning: Models, methods, and a recipe*, in "SIAM Journal on Scientific Computing", 2010, To appear US TR .

International Peer-Reviewed Conference/Proceedings

[32] K. AGRAWAL, A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *Mapping filtering streaming applications with communication costs*, in "21st ACM Symposium on Parallelism in Algorithms and Architectures SPAA 2009", ACM Press, 2009 US .

[33] K. AGRAWAL, A. BENOIT, L. MAGNAN, Y. ROBERT. *Scheduling algorithms for linear workflow optimization*, in "IPDPS'2010, the 24th IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2010, To appear.

[34] M. ALDINUCCI, H. L. BOUZIANE, M. DANELUTTO, C. PÉREZ. *STKM on SCA: a Unified Framework with Components, Workflows and Algorthmic Skeletons*, in "15th International European Conference on Parallel and Distributed Computing (Euro-Par 2009), Delft, Netherlands", LNCS, vol. 5704, Springer, August 2009, p. 678 – 690 IT .

[35] F. ANDRÉ, G. GAUVRIT, C. PÉREZ. *Dynamic Adaptation of the Master-Worker Paradigm*, in "Proc. of the IEEE 9th International Conference on Computer and Information Technology, Xiamen, China", IEEE Computer Society, October 2009, to appear.

[36] L. BEN SAAD, B. TOURANCHEAU. *Multiple Mobile Sinks Positioning in Wireless Sensor Networks for Buildings*, in "SensorComm", IEEE-IARIA, 2009.

[37] A. BENOIT. *Comparison of Access Policies for Replica Placement in Tree Networks*, in "15th International Euro-Par Conference, Delft, The Netherlands", LNCS, Springer Verlag, August 2009.

[38] A. BENOIT, H. CASANOVA, V. REHN-SONIGO, Y. ROBERT. *Resource allocation for multiple concurrent in-network stream-processing applications*, in "HeteroPar'2009: Seventh Int. Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms, jointly held with Euro-Par 2009", LNCS, Springer Verlag, 2009, To appear. Received the Best Paper Award. US .

[39] A. BENOIT, H. CASANOVA, V. REHN-SONIGO, Y. ROBERT. *Resource allocation strategies for in-network stream processing*, in "11th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2009", IEEE Computer Society Press, 2009 US .

[40] A. BENOIT, A. DOBRILA, J.-M. NICOD, L. PHILIPPE. *Throughput optimization for micro-factories subject to failures*, in "ISPDC'2009, 8th International Symposium on Parallel and Distributed Computing, Lisbon, Portugal", July 2009, p. 11-18.

[41] A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *Filter placement on a pipelined architecture*, in "11th Workshop on Advances in Parallel and Distributed Computational Models APDCM 2009", IEEE Computer Society Press, 2009.

[42] A. BENOIT, F. DUFOSSÉ, Y. ROBERT. *On the complexity of mapping pipelined filtering services on heterogeneous platforms*, in "IPDPS'2009, the 23rd IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2009.

[43] A. BENOIT, B. GAUJAL, M. GALLET, Y. ROBERT. *Computing the throughput of replicated workflows on heterogeneous platforms*, in "ICPP'2009, the 38th International Conference on Parallel Processing", IEEE Computer Society Press, 2009, To appear.

[44] A. BENOIT, M. HAKEM, Y. ROBERT. *Optimizing the latency of streaming applications under throughput and reliability constraints*, in "ICPP'2009, the 38th International Conference on Parallel Processing", IEEE Computer Society Press, 2009, To appear.

[45] A. BENOIT, L. MARCHAL, J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Resource-aware allocation strategies for divisible loads on large-scale systems*, in "18th International Heterogeneity in Computing Workshop HCW 2009", IEEE Computer Society Press, 2009.

[46] A. BENOIT, P. RENAUD-GOUD, Y. ROBERT. *Performance and energy optimization of concurrent pipelined applications*, in "IPDPS'2010, the 24th IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2010, To appear.

[47] A. BENOIT, Y. ROBERT, A. ROSENBERG, F. VIVIEN. *Static strategies for worksharing with unrecoverable interruptions*, in "IPDPS'2009, the 23rd IEEE International Parallel and Distributed Processing Symposium", IEEE Computer Society Press, 2009 US .

[48] A. BENOIT, Y. ROBERT, A. ROSENBERG, F. VIVIEN. *Static worksharing strategies for heterogeneous computers with unrecoverable failures*, in "HeteroPar'2009: Seventh Int. Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms, jointly held with Euro-Par 2009", LNCS, Springer Verlag, 2009, To appear US .

[49] J. BIGOT, C. PÉREZ. *Increasing Reuse in Component Models through Genericity*, in "Proc of the 11th International Conference on Software Reuse, Falls Church, Virginia, USA", LNCS, Springer Verlag, October 2009, To appear.

[50] G. CAILLAT, O. LODYGENSKY, G. FEDAK, H. HE, Z. BALATON, Z. FARKAS, G. GOMBAS, P. KACSUK, R. LOVAS, A. C. MAROSI, I. KELLEY, I. TAYLOR, G. TERSTYANSZKY, T. KISS, M. CARDENAS-MONTES, A. EMMEN, F. ARAUJO. *EDGeS: The art of bridging EGEE to BOINC and XtremWeb*, in "Proceedings of Computing in High Energy and Nuclear Physics (CHEP'09) (Abstract), Prague, Czech Republic", March 2009 HU GB PT ES NL .

[51] Y. CANIOU, E. CARON, G. CHARRIER, F. DESPREZ. *Meta-Scheduling and Task Reallocation in a Grid Environment*, in "The Third IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2009), Sliema, Malta", October 2009, p. 6-6.

[52] Y. CANIOU, J.-S. GAY. *Simbatch: an API for simulating and predicting the performance of parallel resources managed by batch systems*, in "Workshop on Secure, Trusted, Manageable and Controllable Grid Services (SGS), held in conjunction with EuroPar'08", LNCS, vol. 5415, 2009, p. 223–234.

[53] E. CARON, A. DATTA, B. DEPARDON, L. LARMORE. *A Self-Stabilizing k-Clustering Algorithm Using an Arbitrary Metric*, in "Euro-Par 2009, Delft, The Netherlands", vol. LNCS 5704, TU Delft, August 25-28 2009, p. 602-614.

[54] E. CARON, F. DESPREZ, D. LOUREIRO, A. MURESAN. *Cloud Computing Resource Management through a Grid Middleware: A Case Study with DIET and Eucalyptus*, in "IEEE International Conference on Cloud Computing (CLOUD 2009), Bangalore, India", IEEE, September 2009, To appear in the Work-in-Progress Track from the CLOUD-II 2009 Research Track..

[55] E. CARON, C. KLEIN, C. PÉREZ. *Efficient Grid Resource Selection for a CEM Application*, in "RenPar'19. 19ème Rencontres Francophones du Parallélisme, Toulouse, France", September 2009.

[56] F. CARRIER, S. DEVISMES, F. PETIT, Y. RIVIERRE. *Space-Optimal Deterministic Rendezvous*, in "Second International Workshop on Reliability, Availability, and Security (WRAS 2009), Hiroshima, Japan", IEEE Computer Society, 2009, To appear.

[57] S. DEVISMES, F. PETIT, S. TIXEUIL. *Optimal Probabilistic Ring Exploration by Asynchronous Oblivious Robots*, in "16th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2009), Piran, Slovenia", Lecture Notes in Computer Science, Springer, vol. 5804, 2009, p. 230–241.

[58] S. DIAKITÉ, L. MARCHAL, J.-M. NICOD, L. PHILIPPE. *Steady-State for Batches of Identical Task Graphs*, in "Euro-Par 2009, Delft University of Technology, Delft, the Netherlands", LNCS, vol. 5704, August 2009, p. 203–215.

[59] Y. DIEUDONNÉ, S. DOLEV, F. PETIT, M. SEGAL. *Brief announcement: deaf, dumb, and chatting robots*, in "28th Annual ACM Symposium on Principles of Distributed Computing (PODC 2009), Calgary, Canada", ACM, 2009, p. 308-309 IL .

[60] Y. DIEUDONNÉ, S. DOLEV, F. PETIT, M. SEGAL. *Deaf, Dumb, and Chatting Robots, Enabling Distributed Computation and Fault-Tolerance Among Stigmergic Robots*, in "13th International Conference On Principles of Distributed Systems (OPODIS 2009), Nîmes, France", Lecture Notes in Computer Science, vol. 5923, Springer, 2009, p. 71-85 IL .

[61] G. FEDAK. *Recent Advances and Research Challenges in Desktop Grid and Volunteer Computing*, in "Proceedings of the EuroPAR 2009 Workshops, CoreGrid ERCIM Working Group Workshop on Grids, P2P and Service Computing, Delft, Netherlands", LNCS, Aug 2009.

[62] M. GALLET, L. MARCHAL, F. VIVIEN. *Efficient Scheduling of Task Graph Collections on Heterogeneous Resources*, in "International Parallel and Distributed Processing Symposium IPDPS'2009", IEEE Computer Society Press, 2009.

[63] Y. GU, Q. WU, A. BENOIT, Y. ROBERT. *Complexity analysis and algorithmic development for pipeline mappings in heterogeneous networks*, in "28th ACM Symposium on Principles of Distributed Computing PODC 2009", ACM Press, 2009, Short communication. US .

[64] Y. GU, Q. WU, A. BENOIT, Y. ROBERT. *Optimizing end-to-end performance of distributed applications with linear computing pipelines*, in "ICPADS'2009, the 15th International Conference on Parallel and Distributed Systems", IEEE Computer Society Press, 2009 US .

[65] H. HE, G. FEDAK, B. TRAN, F. CAPPELLO. *BLAST Application with Data-aware Desktop Grid Middleware*, in "Proceedings of 9th IEEE International Symposium on Cluster Computing and the Grid CCGRID'09, Shanghai, China", May 2009, p. 284–291 CN .

[66] M. JACQUELIN, L. MARCHAL, Y. ROBERT. *Complexity analysis and performance evaluation of matrix product on multicore architectures*, in "ICPP'2009, the 38th International Conference on Parallel Processing", IEEE Computer Society Press, 2009, To appear.

[67] A. C. MAROSI, P. KACSUK, G. FEDAK, O. LODYGENSKY. *Sandboxing for Desktop Grids using virtualization*, in "Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing PDP 2010, Pisa, Italy", February 2010 HU .

[68] Y. MAZZER, B. TOURANCHEAU. *Comparisons for 6LoWPAN Implementation on Wireless Sensor Networks*, in "SensorComm", IEEE-IARIA, 2009.

[69] J.-F. PINEAU, Y. ROBERT, F. VIVIEN. *Energy-aware scheduling of flow applications on master-worker platforms*, in "Euro-Par 2009 - Parallel Processing", LNCS, vol. 5704, Springer Verlag, 2009, p. 281-292.

[70] M. STILLWELL, D. SCHANZENBACH, F. VIVIEN, H. CASANOVA. *Resource Allocation using Virtual Clusters*, in "9th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 09)", 2009 US .

[71] M. STILLWELL, F. VIVIEN, H. CASANOVA. *Dynamic Fractional Resource Scheduling for HPC Workloads*, in "24th IEEE International Parallel and Distributed Processing Symposium (IPDPS)", IEEE CS Press, 2010, To appear US .

[72] B. TOURANCHEAU, Y. MAZZER, G. KRAUSS, V. MARVAN, F. KUZNICK. *Software Calibration of Wirelessly Networked Sensors*, in "SensorComm", IEEE-IARIA, 2009.

[73] B. UÇAR, ÜMIT. V. ÇATALYÜREK. *On scalability of hypergraph models for sparse matrix partitioning*, in "Proceedings of PDP 2010: 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing", 2010, to appear US .

National Peer-Reviewed Conference/Proceedings

[74] G. CHARRIER, Y. CANIOU. *Ordonnancement et réallocation de tâches sur une grille de calcul*, in "RenPar'19, 19e Rencontres Francophones du Parallélisme, Toulouse", September 2009.

[75] B. DEPARDON. *Un algorithme auto-stabilisant pour le problème du k-partitionnement sur graphe pondéré*, in "RenPar'19, 19e Rencontres Francophones du Parallélisme, Toulouse", September 2009.

[76] S. DEVISMES, C. DELPORTE-GALLET, H. FAUCONNIER, F. PETIT, S. TOUEG. *Quand le consensus est plus simple que la diffusion fiable*, in "11e rencontres francophones sur les aspects algorithmiques des télécommunications (Algotel 2009), Carry-Le-Rouet, France", 2009, To appear.

[77] S. DEVISMES, F. PETIT, S. TIXEUIL. *Exploration Optimale Probabiliste d'un Anneau par des Robots Asynchrones et Amnésiques*, in "11e rencontres francophones sur les aspects algorithmiques des télécommunications (Algotel 2009), Carry-Le-Rouet, France", 2009, To appear.

[78] Y. DIEUDONNÉ, F. PETIT. *Squaring the Circle with Weak Mobile Robots*, in "11e rencontres francophones sur les aspects algorithmiques des télécommunications (Algotel 2009), Carry-Le-Rouet, France", 2009, To appear.

[79] A. DOBRILA. *Optimisation du débit dans les micro-usines sujettes aux pannes liées aux tâches et aux machines*, in "RenPar'19, 19éme Rencontres Francophones du Parallélisme, Toulouse", September 2009.

Scientific Books (or Scientific Book chapters)

[80] O. BEAUMONT, L. MARCHAL. *Steady-state scheduling*, in "Introduction to Scheduling", Chapman and Hall/CRC Press, 2009, To appear.

[81] A. BENOIT, Y. ROBERT. *Multi-criteria mapping techniques for pipeline workflows on heterogeneous platforms*, in "Recent Developments in Grid Technology and Applications", G. GRAVVANIS (editor), Nova Science Publishers, 2009, p. 65-99.

[82] V. BRETON, E. CARON, F. DESPREZ, G. LE MAHEC. *High Performance BLAST Over the Grid*, in "Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine and Healthcare", IGI Global, 2009.

[83] Y. CANIOU, E. CARON, F. DESPREZ, H. NAKADA, K. SEYMOUR, Y. TANAKA. *High performance GridRPC middleware*, in "Grid Technology and Applications: Recent Developments", G. GRAVVANIS, J. MORRISON, H. ARABNIA, D. POWER (editors), Nova Science Publishers, 2009, At Prepress. Pub. Date: 2009, 2nd quarter. ISBN 978-1-60692-768-7.

[84] F. CAPPELLO, G. FEDAK, D. KONDO, P. MALÉCOT, A. REZMERITA. *Chapter 3: Desktop Grids: From Volunteer Distributed Computing to High Throughput Computing Production Platforms*, in "Handbook of Research on Scalable Computing Technologies", K.-C. LI, C.-H. HSU, L. T. YANG, J. DONGARRA, H. ZIMA (editors), IGI Global, July 2009, p. 31–61.

[85] E. CARON, F. DESPREZ, F. PETIT, C. TEDESCHI. *DLPT: A P2P tool for Service Discovery in Grid Computing*, in "Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications", N. ANTONOPOULOS, G. EXARCHAKOS, M. LI, A. LIOTTA (editors), IGI Global, December 2009, Released: December 2009. ISBN-13: 978-1615206865..

[86] M. GALLET, Y. ROBERT, F. VIVIEN. *Divisible load scheduling*, in "Introduction to Scheduling", Chapman and Hall/CRC Press, 2009.

[87] Y. ROBERT, F. VIVIEN. *Algorithmic Issues in Grid Computing*, in "Algorithms and Theory of Computation Handbook", Chapman and Hall/CRC Press, 2009.

Books or Proceedings Editing

[88] J. DONGARRA, B. TOURANCHEAU (editors). *Cluster and Computational Grids for Scientific Computing*, Parallel Processing Letters, 2009 US .

[89] Y. ROBERT (editor). *Special issue on IPDPS'2008*, J. Parallel and Distributed Computing 69, 9, 2009.

[90] Y. ROBERT, F. VIVIEN (editors). *Introduction to Scheduling*, Chapman and Hall/CRC Press, 2009.

Other Publications

[91] P. R. AMESTOY, I. S. DUFF, D. RUIZ, B. UÇAR. *Towards parallel bipartite matching algorithms*, May 2009, Presentation at Scheduling for large-scale systems, Knoxville, Tennessee, USA GB .

[92] P. R. AMESTOY, I. S. DUFF, TZ. SLAVOVA, B. UÇAR. *Out-of-core solution for singleton rhs vectors*, March 2009, Presentation at SIAM Conference on Computational Science and Engineering (CSE09), Miami, Florida, USA GB .

[93] P. R. AMESTOY, I. S. DUFF, TZ. SLAVOVA, B. UÇAR. *Out-of-core solution for singleton right hand-side vectors*, February 2009, Poster at Dagstuhl Seminar on Combinatorial Scientific Computing GB .

[94] I. S. DUFF, B. UÇAR. *Combinatorial problems in solving linear systems*, February 2009, Invited presentation at Dagstuhl Seminar on Combinatorial Scientific Computing delivered by Iain S. Duff GB .

[95] ÜMIT. V. ÇATALYÜREK, B. UÇAR. *Partitioning sparse matrices*, March 2009, Presentation at SIAM Conference on Computational Science and Engineering (CSE09), Miami, Florida, USA delivered by Ümit V. Çatalyürek US .

References in notes

[96] R. BUYYA (editor). *High Performance Cluster Computing*, vol. 2: Programming and Applications, Prentice Hall, 1999, ISBN 0-13-013784-7.

[97] P. CHRÉTIENNE, E. G. COFFMAN JR., J. K. LENSTRA, Z. LIU (editors). *Scheduling Theory and its Applications*, John Wiley and Sons, 1995.

[98] I. FOSTER, C. KESSELMAN (editors). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, 1998.

[99] A. ORAM (editor). *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology*, O'Reilly, 2001.

[100] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT. *Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers*, in "Comput. Methods Appl. Mech. Eng.", vol. 184, 2000, p. 501–520.

[101] D. ARNOLD, S. AGRAWAL, S. BLACKFORD, J. DONGARRA, M. MILLER, K. SAGI, Z. SHI, S. VADHIYAR. *Users' Guide to NetSolve V1.4*, n° CS-01-467, University of Tennessee, Knoxville, TN, July 2001, http://www.cs.utk.edu/netsolve/, Computer Science Dept. Technical Report.

[102] M. BAKER. *Cluster Computing White Paper*, 2000.

[103] E. CARON, A. CHIS, F. DESPREZ, A. SU. *Plug-in Scheduler Design for a Distributed Grid Environment*, in "4th International Workshop on Middleware for Grid Computing - MGC 2006, Melbourne, Australia", November 27th 2006, In conjunction with ACM/IFIP/USENIX 7th International Middleware Conference 2006.

[104] I. S. DUFF, J. K. REID. *The Multifrontal Solution of Indefinite Sparse Symmetric Linear Systems*, in "ACM Transactions on Mathematical Software", vol. 9, 1983, p. 302-325.

[105] I. S. DUFF, J. K. REID. *The Multifrontal Solution of Unsymmetric Sets of Linear Systems*, in "SIAM Journal on Scientific and Statistical Computing", vol. 5, 1984, p. 633-641.

[106] H. EL-REWINI, H. H. ALI, T. G. LEWIS. *Task Scheduling in Multiprocessing Systems*, in "Computer", vol. 28, n^{o} 12, 1995, p. 27–37.

[107] G. FEDAK, C. GERMAIN, V. NÉRI, F. CAPPELLO. *XtremWeb : A Generic Global Computing System*, in "CCGRID2001, workshop on Global Computing on Personal Devices", IEEE Press, May 2001.

[108] M. FERRIS, M. MESNIER, J. MORÉ. *NEOS and Condor: Solving Optimization Problems Over the Internet*, in "ACM Transactions on Mathematical Sofware", vol. 26, n^{o} 1, 2000, p. 1-18, http://softlib.rice.edu/pub/CRPC-TRs/reports/CRPC-TR98763-S.pdf.

[109] C. GERMAIN, G. FEDAK, V. NÉRI, F. CAPPELLO. *Global Computing Systems*, in "Lecture Notes in Computer Science", vol. 2179, 2001, p. 218–227.

[110] J. W. H. LIU. *The Role of Elimination Trees in Sparse Factorization*, in "SIAM Journal on Matrix Analysis and Applications", vol. 11, 1990, p. 134–172.

[111] S. MATSUOKA, H. NAKADA, M. SATO, S. SEKIGUCHI. *Design Issues of Network Enabled Server Systems for the Grid*, 2000, Grid Forum, Advanced Programming Models Working Group whitepaper.

[112] H. NAKADA, S. MATSUOKA, K. SEYMOUR, J. DONGARRA, C. LEE, H. CASANOVA. *GridRPC: A Remote Procedure Call API for Grid Computing*, in "Grid 2002, Workshop on Grid Computing, Baltimore, MD, USA", Lecture Notes in Computer Science, n^{o} 2536, November 2002, p. 274-278.

[113] H. NAKADA, M. SATO, S. SEKIGUCHI. *Design and Implementations of Ninf: towards a Global Computing Infrastructure*, in "Future Generation Computing Systems, Metacomputing Issue", vol. 15, n^{o} 5-6, 1999, p. 649-658.

[114] M. G. NORMAN, P. THANISCH. *Models of Machines and Computation for Mapping in Multicomputers*, in "ACM Computing Surveys", vol. 25, n^{o} 3, 1993, p. 103–117.

[115] M. SATO, M. HIRANO, Y. TANAKA, S. SEKIGUCHI. *OmniRPC: A Grid RPC Facility for Cluster and Global Computing in OpenMP*, in "Lecture Notes in Computer Science", vol. 2104, 2001, p. 130–136.

[116] B. A. SHIRAZI, A. R. HURSON, K. M. KAVI. *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Science Press, 1995.

[117] R. WOLSKI, N. T. SPRING, J. HAYES. *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing*, in "Future Generation Computing Systems, Metacomputing Issue", vol. 15, n^{o} 5–6, October 1999, p. 757–768.